Universidad Carlos III de Madrid

Departamento de Teoría de la Señal y Comunicaciones

Detección automática de paráfrasis sobre un corpus de preguntas en inglés

Trabajo de Fin de Grado

Autor:

Iñaki Renedo Muñoz de la Peña

Tutor:

Jesús Cid Sueiro

Grado en Ingeniería de Sistemas Audiovisuales

Escuela Politécnica Superior Universidad Carlos III de Madrid

Leganés, junio de 2018

Resumen

El aumento exponencial de la información escrita durante los últimos años ha creado la necesidad de desarrollar herramientas con el objetivo de procesar de manera automática todo este conocimiento. Existen gran cantidad de aplicaciones y técnicas implementadas con el fin de procesar automáticamente la información escrita. Una de las ramas de investigación más popular a causa de su amplia aplicabilidad es la detección de paráfrasis.

En el presente Trabajo de Fin de Grado, se presenta la solución a un problema de detección de paráfrasis en textos cortos. Concretamente, se trata de un problema de detección de preguntas repetidas sobre un corpus de pares de preguntas en inglés. Con el objetivo de solucionar un problema de estas características, se han combinado varias técnicas basadas en la similitud léxica y la semántica de las palabras.

En el presente Trabajo de Fin de Grado, se revisa el estado de las investigaciones sobre la detección de paráfrasis y se describen las técnicas más destacadas. Las técnicas basadas en aprendizaje automático son las que presentan mejores prestaciones, sin embargo, el problema de detección de paráfrasis en textos cortos no ha sido resuelto aún con carácter definitivo.

Abstract

The exponential increase in written information over the last few years has created the need to develop tools with the aim of automatically processing all this knowledge. There are many applications and techniques implemented in order to automatically process written information. One of the most popular research branches is paraphrase detection because the amount of uses that it has.

In this final degree project, is presented a paraphrase detection problem in short texts. Specifically, it is a repeated questions detection problem on a corpus of pairs of questions in english. In order to solve a problem of these characteristics, several techniques based on lexical and semantic similarity of words have been combined.

In addition, in this final degree project, the status of the researches in paraphrase detection is reviewed and the most outstanding techniques are described. The techniques based on machine learning are those that present better performance, however, the problem of paraphrase detection in short texts has not been solved definitely.

Índice general

Índice de tablas……..…..4

Índice de figuras……5

Índice de ecuaciones……6

1. Introducción……..…….8
   - 1.1. Situación actual..…….....8
   - 1.2. Problema planteado…….....9
   - 1.3. Motivación: aplicaciones y dificultades……...…..10
   - 1.4. Organización del trabajo…….…..12

2. Estado del arte de la detección de paráfrasis……..…..14
   - 2.1. Introducción a la detección de paráfrasis……...…..14
   - 2.2. Técnicas para la detección de paráfrasis……...…..15
     - 2.2.1. Técnicas basadas en la similitud de las palabras……....15
       - Técnicas basadas en la similitud léxica de las palabras……...17
       - Técnicas basadas en la similitud semántica de las palabras……...18
         1. Método de detección de paráfrasis con métricas de similitud semántica basadas en corpus y conocimiento…….....22
         2. Método de detección de paráfrasis a partir de la matriz de similitud……..……...23
     - 2.2.2. Métodos basados en aprendizaje automático…….....24
       1. Método de detección de paráfrasis por disimilitud…….....25
       2. Método de detección de paráfrasis por canonicalización de textos……..……...26
       3. Método de detección de paráfrasis combinando medidas de similitud…….……27
       4. Método de detección de paráfrasis mediante métricas de traducción automática……...28
   - 2.3. Conclusiones……..28

3. Problema planteado y solución propuesta……..…..31
   - 3.1. Problema planteado……..31
   - 3.2. Enfoque de la solución…….32
     - 3.2.1. Pre-procesado del texto……...33
     - 3.2.2. Evaluación……..34

1. Método 1. Comparador literal de palabras……...35
2. Método 2. Método de conteo……...35
3. Método 3. Comparador semántico de palabras……37
4. Método 4. Máxima similitud……..38
5. Método 5. Método híbrido……39

3.2.3. Decisión……...41

3.3. Tecnologías utilizadas……..……...42

4. Experimentos y resultados……...45

4.1. Colección de datos……..….45

4.2. Experimentos y resultados……...58

- Experimento 1. Métodos basados en la repetición de palabras: Comparador literal de palabras y Método de conteo……...58
- Experimento 2. Métodos basados en la similitud semántica de palabras: Comparador semántico de palabras y Máxima similitud……...59
- Experimento 3. Método que combina medidas de similitud léxica y medidas de similitud semántica: Método híbrido……..61

5. Marco regulador……....63

5.1. Legislación aplicable……..……....63

5.2. Estándares técnicos……..……..……63

5.3. Propiedad intelectual……..……....63

6. Entorno Socio-económico……64

6.1. Presupuesto……...64

6.2. Impacto socio-económico……....65

7. Conclusiones y trabajo futuro…….……66

7.1. Conclusiones……..66

7.2. Trabajo a futuro……..66

Anexo A. Resumen en inglés……...……69

Bibliografía……..…….....78

Índice de tablas

Tabla 2.1. Matriz de confusión……16
Tabla 3.1. Ejemplo Método de conteo……37
Tabla 4.1. Estadísticas sobre la distribución de palabras en el conjunto de datos sin preprocesar……46
Tabla 4.2. Estadísticas sobre la repetición de palabras entre los pares del conjunto de datos sin preprocesar……47
Tabla 4.3. Estadísticas sobre la distribución de palabras en el conjunto de datos preprocesado……52
Tabla 4.4. Estadísticas sobre la repetición de palabras entre los pares del conjunto de datos preprocesado……53
Tabla 4.5.. Exactitud y Medida F para el Método 1. Comparador literal de palabras y Método 2. Método de Conteo……59
Tabla 4.6.. Exactitud y Medida F para el Método 3. Comparador semántico de palabras y Método 4. Máxima similitud……60
Tabla 4.7. Exactitud y Medida F para el Método 5. Método híbrido y para la media aritmética de los cuatro primeros métodos……61
Tabla 6.1. Presupuesto de los recursos utilizados para la ejecución del presente Trabajo de Fin de Grado……64
Chart A.1. Accuracy and F measure for the methods implemented in this final degree project……76

Índice de figuras

Figura 2.1. Extracto de la Jerarquía “es un” de WordNet …...……19
Figura 2.2. Esquema general de los métodos de aprendizaje automático orientados a la detección de paráfrasis …...……..25
Figura 3.1. Diagrama de bloques de las fases de la solución…….33
Figura 4.1. Histograma de aparición de las preguntas del corpus de datos completo……...47
Figura 4.2. Histograma de aparición de las preguntas del corpus de entrenamiento…….48
Figura 4.3. Histograma de aparición de las preguntas del corpus de prueba……..48
Figura 4.4. Histograma de las palabras más frecuentes del corpus de datos completo…...54
Figura 4.5. Histograma de las palabras más frecuentes del corpus de entrenamiento……..54
Figura 4.6. Histograma de las palabras más frecuentes del corpus de prueba……...55
Figure A.1. General scheme of machine learning algorithms operation……...72
Figure A.2. General scheme of the solution process……...73

Índice de ecuaciones

Ecuación 2.1. Ecuación de la precisión positiva……..16
Ecuación 2.2. Ecuación de la precisión negativa……..……..16
Ecuación 2.3. Ecuación del recuerdo positivo……...…..16
Ecuación 2.4. Ecuación del recuerdo negativo……...16
Ecuación 2.5. Ecuación de la medida F positiva ……...……16
Ecuación 2.6. Ecuación de la medida F negativa ...……...16
Ecuación 2.7. Ecuación de la medida F……..……..16
Ecuación 2.8. Ecuación de la exactitud……..……..16
Ecuación 2.9. Ecuación del coeficiente de Dice……...……..17
Ecuación 2.10. Ecuación del coeficiente de Jaccard……..17
Ecuación 2.11. Ecuación del coeficiente del traslape……...……..17
Ecuación 2.12. Ecuación del coeficiente del coseno……..18
Ecuación 2.13. Ecuación de la métrica Leacock & Chodorow ……..20
Ecuación 2.14. Ecuación del contenido informativo (IC) ……..……...…..20
Ecuación 2.15. Ecuación de la métrica Wu & Palmer……21
Ecuación 2.16. Ecuación de la métrica Resnik ..…….21
Ecuación 2.17. Ecuación de la métrica Lin..…….21
Ecuación 2.18. Ecuación de la métrica Jiang y Conrath..……...……21
Ecuación 2.19. Ecuación de la métrica Path length..……..……21
Ecuación 2.20. Ecuación de la similitud semántica del método desarrollado por Mihalcea, R., Corley, C. & Strapparava, C. [52]..……...…22
Ecuación 2.21. Ecuación de la medida de T_1 utilizada en la ecuación 2.20……...….22
Ecuación 2.22. Ecuación de la medida de T_2 utilizada en la ecuación 2.20……....22
Ecuación 2.23. Ecuación de la Frecuencia inversa de documento..…….22
Ecuación 2.24. Ecuación del recuerdo de dependencia de S_1 ……...27
Ecuación 2.25. Ecuación del recuerdo de dependencia de S_2 ……...27
Ecuación 2.26. Ecuación de la medida F del recuerdo de dependencia de S_1 y S_2 ……..27
Ecuación 3.1. Ecuación de la media aritmética de los cuatro primero métodos (1-4).....34
Ecuación 3.2. Ecuación de la medida de similitud semántica del Método 1. Comparador literal de palabras……...35
Ecuación 3.3. Ecuación de normalización de la medida de similitud semántica del Método 3. Comparador semántico de palabras para una métrica de similitud dada……..38
Ecuación 3.4. Ecuación de la medida de similitud semántica del Método 1. Comparador semántico de palabras.……..38
Ecuación 3.5. Ecuación de la normalización del valor de similitud semántica del Método 4. Máxima similitud, para una métrica dada.……...39
Ecuación 3.6. Ecuación de la medida de similitud semántica del Método 4. Máxima similitud……....39
Ecuación 3.7. Ecuación de la media aritmética para obtener un solo valor relacionado con la similitud semántica en el Método 5. Método híbrido..…….41
Ecuación 3.8. Ecuación de la medida de similitud semántica del Método 5. Método híbrido.……...41

Equation A.1. Equation of the semantic similarity value normalization of the Method 3. Semantic word comparator to a given semantic similarity measure……...74
Equation A.2. Equation of the semantic similarity value normalization of the Method 4. Maximum similarity to a given semantic similarity measure…….75

1. Introducción

1.1. Situación actual

En los últimos tiempos, la cantidad de información escrita en formato digital ha experimentado un crecimiento exponencial. Gracias a Internet y a las nuevas tecnologías esta gran cantidad de textos está al alcance de cualquier persona que disponga de un dispositivo electrónico adecuado.

Sin embargo, debido al inmenso volumen de información escrita, todo el conocimiento que éste alberga es totalmente inútil si no se disponen de las herramientas adecuadas para manejar toda esta información. Por ello, surge la necesidad de procesar los textos de manera automática, lo que facilitará múltiples tareas.

Para el procesado automático de la información textual se ha recurrido al Procesamiento del Lenguaje Natural (PNL) [1], una rama de investigación de la Inteligencia Artificial.

El objetivo principal de esta rama consiste en facilitar el acceso y la organización de la información. Estos objetivos son llevados a cabo mediante métodos y técnicas computacionales que se encargan de realizar análisis precisos de la información escrita [2] [3].

Con el fin de llevar a cabo este objetivo principal, el Procesamiento del Lenguaje Natural ha implementado otros sistemas con unos objetivos secundarios. Estos sistemas se encargan, por ejemplo, de agrupación de documentos [4], clasificación de textos [5], detección de plagios o semejanza entre documentos [6]. Estos sistemas tienen en común que todos utilizan para su funcionamiento la detección de similitud textual.

La detección de similitud textual es una de las tareas más importantes y complicadas dentro del procesamiento automático de textos. Es muy importante por la cantidad de aplicaciones que tiene y es complicada por la complejidad y la subjetividad del lenguaje.

En los últimos años se han llevado a cabo con éxito múltiples investigaciones sobre la detección automática de similitud textual. La mayoría de las líneas de investigación se centran en la detección automática de similitud de textos largos, en los que se dispone de información suficiente con la que calcular valores de similitud notablemente fiables. Por lo tanto, este problema ya está relativamente resuelto [7].

No obstante, cuando se trata de calcular la similitud entre dos textos cortos, como dos oraciones, la dificultad aumenta y la calidad de las medidas de similitud disminuyen sustancialmente ya que la información que aportan los textos no es suficiente.

En estos casos los métodos tradicionales que utilizan la frecuencia de palabras no funcionan y es necesario recurrir a técnicas más complejas para obtener medidas más fiables [8].

Estas técnicas enfocadas al cálculo de similitud en textos cortos son muy variadas. Algunos proyectos se basan, entre otras muchas técnicas, en la comparación de palabras [9], en técnicas de detección de secuencias de palabras [10] o en la comparación de medidas de similitud semántica [11]. Otras técnicas que han presentado muy buenos resultados son las basadas en aprendizaje automático [12][13].

Recientemente, se ha demostrado que los métodos que combinan varias técnicas [14], de las citadas anteriormente u otras no nombradas, alcanzan resultados mucho más precisos y fiables.

1.2. Problema Planteado

En el presente trabajo se plantea un problema de detección de preguntas repetidas. Este problema está planteado en un reto de Kaggle (www.kaggle.com), una plataforma que organiza competiciones relacionadas con “machine learning”. Este reto de la plataforma Kaggle (https://www.kaggle.com/c/quora-question-pairs), es una muestra representativa del interés tecnológico del problema.

De esta plataforma (www.kaggle.com) se obtuvo la colección de datos necesaria para todo el desarrollo de la solución del problema. Esta colección de datos consiste en un conjunto de 404.290 pares de preguntas en inglés etiquetados previamente, es decir, los pares de preguntas han sido previamente identificados según la equivalencia semántica de sus preguntas. Por lo tanto, se ha llevado a cabo una clasificación binaria previa en la que se han identificado los pares como: ‘equivalentes’ o ‘no equivalentes’.

El objetivo del problema consiste en decidir si las preguntas de cada par significan lo mismo o no. Más concretamente, se trata de un problema de detección de paráfrasis entre las preguntas de cada par del conjunto mencionado anteriormente, ya que el objetivo del problema es decidir en cada par si las preguntas están preguntando lo mismo pero con distintas palabras, estructura y/o estilo o no.

Por lo tanto el objetivo principal del trabajo consiste en decidir de la manera más precisa posible, si cada par de preguntas está formado por dos preguntas equivalentes semánticamente o, por el contrario, está formado por dos preguntas no equivalentes. Con el fin de resolver este objetivo principal, se ha segmentado la solución del problema en tres fases cada una con un objetivo secundario: pre-procesado del texto, evaluación de la similitud semántica textual y decisión final. Estos objetivos secundarios es preciso llevarlos a cabo correctamente de forma secuencial para poder obtener una solución final fiable.

En primer lugar, será necesario llevar a cabo un buen pre-procesado del texto que lo adecúe convenientemente para facilitar y optimizar al máximo el cálculo del valor que representará la similitud semántica textual entre las preguntas de cada par. Para esto, mediante varios métodos, se segmentan las preguntas en “tokens” (unidades equivalentes a palabras) y se eliminarán aquellos que no aporten información semántica, las denominadas “stopwords” (artículos, preposiciones, pronombres, determinantes, etc.). Además, las

palabras que estén en una forma flexionada se reducirán a su lema (raíz) mediante un proceso llamado lematización. De esta manera podremos extraer más fácilmente la información relevante de cada pregunta.

Posteriormente se evaluarán los datos resultantes y se procederá a calcular un valor entre 0 y 1 (“score”) para cada par de preguntas que representará la similitud semántica textual que existe entre ambas preguntas. Para esto se utilizarán diferentes métodos que serán descritos más adelante y que, mediante varias técnicas calculan un valor representativo de la similitud semántica textual entre las dos preguntas de cada par. Entre estas técnicas podemos encontrar la comparación literal de palabras y la comparación semántica de palabras principalmente. Además, todas las técnicas combinan distintas formas de normalizar los resultados para alcanzar una solución óptima.

Finalmente se deberá decidir, en base al valor (“score”) calculado en la etapa anterior, si cada par de preguntas es equivalente o no. Para esto deberemos obtener un umbral de decisión que nos permita decidir de la mejor manera posible entre los pares de preguntas equivalentes y los que no lo son. El cálculo correcto de este umbral de decisión es de suma importancia ya que nos permitirá optimizar el resultado del método implementado. Su cálculo se describe, con detalle, más adelante.

1.3. Motivación: aplicaciones y dificultades

La necesidad creciente de procesar información de manera automática mencionada anteriormente y las dificultades que implica la detección de similitud semántica en textos cortos causadas por escasa información semántica, la subjetividad del lenguaje o la complejidad de éste, hacen del problema planteado de detección de preguntas equivalentes un problema complejo de resolver y de un carácter necesario en la actualidad. Debido a estas dificultades, para alcanzar una solución del problema efectiva, será necesario combinar varias técnicas de distintas naturalezas que nos permitan extraer la información de las preguntas dadas, de la manera más precisa posible.

A continuación, podemos ver dos ejemplos de pares de la colección de datos utilizada para elaborar la solución del problema que ya están etiquetados según la equivalencia semántica de sus preguntas (“0” si las preguntas no son equivalentes y “1” si las preguntas son equivalentes). En ellos se puede apreciar la complejidad del lenguaje y la necesidad de incluir técnicas más sofisticadas que las técnicas basadas en la búsqueda de palabras repetidas para conseguir calcular un ‘score’ de similitud semántica textual fiable en el ámbito de los textos cortos:

- *Ejemplo 1. Par 1º de la colección de datos:*
    - *"What is the step by step guide to invest in share market in india?"*
    - *"What is the step by step guide to invest in share market?"*

  Evaluado como: "0".

En este caso, podemos apreciar que ambas preguntas comparten la mayoría de las palabras. De hecho todas las palabras de la pregunta 2 están contenidas en la pregunta 1. Sin embargo las preguntas de este par, no son equivalentes semánticamente.

- *Ejemplo 2. Par 32º de la colección de datos:*
    - *"What are some special cares for someone with a nose that gets stuffy during the night?"*
    - *"How can I keep my nose from getting stuffy at night?"*

  Evaluado como: "1".

En este otro caso, podemos apreciar que ambas preguntas comparten sólo algunas palabras (“nose”, “stuffy”, “night”) pero no la mayoría de ellas. Sin embargo ambas preguntas son equivalentes semánticamente.

Por lo tanto, en estos dos casos un método que se base en la comparación literal de palabras para medir la similitud semántica erraría, ya que como podemos observar en estos ejemplos el hecho de compartir muchas palabras o de no hacerlo no garantiza o descarta respectivamente una equivalencia semántica.

Actualmente, el problema del cálculo automático de similitud semántica textual en textos cortos no ha sido resuelto de una manera definitiva. Si bien es cierto que ha habido numerosos trabajos enfocados en este tema [15], ninguno de ellos ha logrado aportar una solución con carácter definitivo. Este hecho, pone de manifiesto el interés por investigar y comparar distintos trabajos centrados en este objetivo o en objetivos relacionados con la finalidad de encontrar una solución fiable al problema propuesto.

El reto de Kaggle también ha supuesto un motivo para investigar y resolver este problema ya que las ventajas que ofrecía esta plataforma han facilitado enormemente el trabajo de investigación. Las principales ventajas que ofrecía Kaggle han sido: la completa y bien etiquetada colección de datos (404.290 pares de preguntas en inglés), en la que se identifican claramente los pares de preguntas que contenían preguntas equivalentes entre sí y los pares de preguntas cuyas preguntas no eran equivalentes, el entorno on-line de desarrollo donde se comenzó a trabajar en la solución del problema planteado y varias propuestas para la solución al problema propuesto que otros competidores compartían en la plataforma con el objetivo de alimentar las soluciones de los demás participantes. En estas propuestas se pueden apreciar métodos, técnicas e ideas complementarias que pueden ser de gran ayuda para construir una solución efectiva y analizar el conjunto de datos disponible.

Además del reto de Kaggle, otro motivo que pone de manifiesto el interés científico de investigar y buscar una solución efectiva al problema planteado es la amplia aplicabilidad que tiene una solución que cumpla estas características.

En concreto aplicar la solución para dos aplicaciones que se basan en la detección de paráfrasis: la detección de preguntas repetidas en el Congreso de los Diputados y la detección de tuits repetidos con el objetivo de encontrar plagiadores en Twitter. Estas aplicaciones fueron dos posibilidades a la hora de elegir el problema a resolver en el Trabajo de Fin de Grado, pero finalmente fueron descartadas por la dificultad que suponía conseguir un conjunto de datos sobre el que trabajar y con el que entrenar la solución frente a las facilidades y ventajas que ofrecía el reto de Kaggle.

Sin embargo, la solución implementada, puede ser adaptada para estas dos aplicaciones que fueron descartadas lo que también ha supuesto una motivación extra para investigar y proponer una solución efectiva al problema planteado en el presente Trabajo de Fin de Grado.

Además de estas dos aplicaciones, como se ha señalado anteriormente, una solución eficiente al problema planteado tiene numerosas aplicaciones lo que ha supuesto una gran motivación a la hora de investigar. Aplicaciones como evaluación de traducción automática [16], clasificación de documentos [17], evaluación de resúmenes redactados automáticamente [18], detección de paráfrasis [19] o sugerencia de respuestas automáticas [20].

La solución que se propone en el presente Trabajo de Fin de Grado al problema planteado consiste en una combinación de varias técnicas, algunas ya implementadas anteriormente y otras implementadas en el presente trabajo por primera vez. Todas estas técnicas que componen la solución del problema serán explicadas detalladamente a continuación, a lo largo del trabajo.

1.4 Organización del trabajo

El contenido del presente trabajo está organizado de la siguiente manera:

En el capítulo 2 se presenta el estado del arte y los trabajos más influyentes en el campo de la detección de paráfrasis en textos cortos. Se enfatiza en los trabajos que abordan esta tarea mediante similitud semántica y aprendizaje automático supervisado principalmente.

En el capítulo 3 se describe detalladamente el problema planteado y los métodos propuestos para resolverlo, explicados y justificados. Además, se presentan los algoritmos más importantes que presentan dichos métodos.

En el capítulo 4 se muestran los resultados obtenidos en los experimentos realizados con los métodos propuestos para la resolución del problema planteado, las conclusiones correspondientes a estos experimentos y se incluye un análisis detallado de las características más relevantes del conjunto de datos que ha sido utilizado para desarrollar la solución del problema planteado.

En el capítulo 5 se presenta el marco regulador del presente Trabajo de Fin de Grado en el que se incluye la legislación aplicable a dicho trabajo, los estándares técnicos que aplican y el estado de la propiedad intelectual en el trabajo.

En el capítulo 6 se hace un análisis del entorno socio-económico en el que se incluye un presupuesto de la elaboración del presente Trabajo de Fin de Grado y un análisis del impacto socio-económico de las aplicaciones que pudiese tener la solución presentada en el presente Trabajo de Fin de Grado.

En el capítulo 7 se presentan las conclusiones extraídas durante la elaboración del presente Trabajo de Fin de Grado, así como posibles líneas de trabajo a futuro.

En el Anexo A se presenta un resumen en inglés del presente Trabajo de Fin de Grado.

En la Bibliografía se presentan las referencias que se han utilizado en el desarrollo del presente Trabajo de Fin de Grado.

2. Estado del arte de la detección de paráfrasis

2.1. Introducción a la detección de paráfrasis

La detección de similitud semántica textual es una rama de investigación muy recurrente durante los últimos años, en torno a la cual se han llevado a cabo un gran número de investigaciones con diferentes enfoques, objetivos y aplicaciones. Uno de los objetivos más frecuentes y útiles para los que puede ser aplicada la detección de similitud semántica textual es la detección de paráfrasis.

Para diseñar un método efectivo que tenga como objetivo la detección de paráfrasis, la primera tarea es definir el concepto de paráfrasis, ya que de otra manera no habrá un rumbo que seguir durante el transcurso de la investigación y será sencillo que la investigación y sus objetivos se distorsionen.

La paráfrasis se define como: *"Frase que, imitando en su estructura otra conocida, se formula con palabras diferentes"* [21]. Es decir, la detección de paráfrasis consiste en detectar si dos fragmentos de texto, pese a no compartir exactamente las mismas palabras, expresan el mismo mensaje. A continuación se presenta un ejemplo de paráfrasis:

1. *"El coche rojo fue el más veloz de la carrera de Malasia"*
2. *"El Ferrari quedó campeón en el gran premio de Malasia"*

En este ejemplo se puede observar que ambas oraciones no comparten las mismas palabras pero el mensaje que expresan es equivalente, por lo que en este caso se puede afirmar que existe paráfrasis.

La detección de paráfrasis puede ser utilizada en numerosas aplicaciones como recuperación de información por medio de similitudes semánticas de búsquedas concretas, traducción automática [22], generación automática de resúmenes [23], identificación de plagio en textos [24] y búsqueda de respuestas [25].

Sin embargo, la detección de paráfrasis también puede ser un fin en sí misma. Puede ser aplicada directamente para detectar preguntas repetidas (equivalentes semánticamente) en el Congreso de los Diputados o para detectar tuits con un mismo mensaje, por ejemplo.

Esta línea de aplicación enfocada a la detección de paráfrasis entre oraciones como objetivo final, ha sido abordada por un gran número de investigadores debido al menor coste computacional relacionado con la corta longitud de los textos a procesar. Estas investigaciones y estudios han dado lugar a varios métodos distintos que a su vez están basados en técnicas de diversas naturalezas.

Las técnicas y los métodos utilizados para la detección de paráfrasis entre oraciones han evolucionado desde sus primeras implementaciones hasta la actualidad. Las primeras técnicas estaban basadas en la similitud léxica de las palabras [26][27][28], es decir, en

función del número de palabras que coincidieran en un par de oraciones o el parecido léxico de las palabras, se decidía si existía paráfrasis o no. Posteriormente, gracias a la herramienta WordNet [29] entre otras, se pudieron agregar a las técnicas anteriores métricas (medidas) que calculaban la similitud semántica entre las palabras de las oraciones a evaluar [30][31].

Además de estas métricas, han sido utilizadas para la detección de paráfrasis otras métricas orientadas a la recuperación de información como la distancia de Manhattan, la distancia euclidiana, similitud del coseno [32] y el uso de modelos probabilísticos [33]. También se han utilizado otras medidas diseñadas específicamente para la paráfrasis, tales como las basadas en n-gramas [34] y medidas asimétricas [35][36].

Recientemente se ha enfocado la tarea de detección de paráfrasis mediante aprendizaje automático supervisado: algoritmos de clasificación como k-vecinos, máquinas de soporte vectorial y máxima entropía [12][13]. Gracias a la combinación de las medidas de similitud semántica con estas técnicas de aprendizaje automático supervisado, se ha logrado aumentar las prestaciones de los métodos que persiguen la detección de paráfrasis [36].

2.2. Técnicas para la detección de paráfrasis.

A continuación se describirán las principales técnicas en las que se basan los métodos cuyo objetivo es lograr una detección de paráfrasis efectiva entre oraciones. No obstante algunos métodos enfocados a detectar paráfrasis en textos más largos, pueden ser fácilmente adaptados para su aplicación con oraciones, por lo que, en las próximas páginas también se incluirán técnicas diseñadas e implementadas para textos de mayor extensión.

2.2.1. Técnicas basadas en la similitud de las palabras

Este tipo de técnicas se basan en la similitud que existe entre las palabras de dos oraciones para calcular un valor que represente la similitud semántica textual y a partir de éste, detectar si existe paráfrasis. Generalmente, el valor calculado que representa la similitud semántica entre dos oraciones, está comprendido entre 0 y 1. Siendo 1 el valor que representa la máxima similitud semántica textual y 0 la discrepancia absoluta.

Estos valores que representan la similitud semántica textual, pueden ser calculados mediante comparaciones léxicas, semánticas o sintácticas de palabras. Para decidir en base a estos valores si una oración es paráfrasis de otra es necesario un valor umbral de decisión. Mediante este valor umbral podremos decidir si en un par de oraciones existe paráfrasis o no. Normalmente si el valor calculado que representa la similitud semántica textual es mayor al valor umbral de decisión, se decidirá que existe paráfrasis, en caso opuesto se decidirá que no hay paráfrasis.

Con el objetivo de evaluar la calidad de los resultados obtenidos es conveniente tener en cuenta los siguientes conjuntos: TP (true positives), TN (true negatives), FP (false positives)

y FN (false negatives). Estos conjuntos se representan en la matriz de confusión que se muestra a continuación:

		PREDICCIÓN	
		Paráfrasis	No paráfrasis
REALIDAD	Paráfrasis	TP	FN
	No paráfrasis	FP	TN

Tabla 2.1. Matriz de confusión

En las columnas se representa la clasificación del método implementado (predicción) mientras que en las filas se representa la clasificación real (realidad), es decir, la clasificación correspondiente a la solución óptima.

En la matriz de confusión podemos observar que TP y TN son los porcentajes de acierto del método implementado (separados en el acierto de los clasificados como positivos y el acierto de los clasificados como negativos) mientras que FP y FN son los porcentajes de error (separados nuevamente en el error al clasificar pares positivos y negativos). A partir de estas cuatro categorías se definen las métricas estándar para evaluar un método de clasificación: precisión (P), recuerdo (R), medida F (F) y exactitud (E). A continuación se presentan las fórmulas de estas métricas de evaluación:

$$P_{positiva} = \frac{TP}{TP + FP} \quad (2.1.)$$

$$P_{negativa} = \frac{TN}{TN + FN} \quad (2.2.)$$

$$R_{positivo} = \frac{TP}{TP + FN} \quad (2.3.)$$

$$R_{negativo} = \frac{TN}{TN + FP} \quad (2.4.)$$

$$F_{positiva} = 2 \times \frac{P_{positiva} R_{positiva}}{P_{positiva} + R_{positiva}} \quad (2.5.)$$

$$F_{negativa} = 2 \times \frac{P_{negativa} R_{negativa}}{P_{negativa} + R_{negativa}} \quad (2.6.)$$

$$F = \frac{F_{positiva} + F_{negativa}}{2} = \frac{TP(2TN+FP+FN) + TN(2TP+FN+FP)}{(2TP+FP+FN)(2TN+FP+FN)} \quad (2.7.)$$

$$E = \frac{TP+TN}{TP + FP+FN+TN} \quad (2.8.)$$

Técnicas basadas en la similitud léxica de las palabras

Este tipo de técnicas fueron las primeras que se desarrollaron para detectar paráfrasis. Se basaban en la comparativa léxica de palabras [26][27][28], es decir, en el número de palabras que compartían ambos fragmentos de texto o su parecido léxico. Estas técnicas no presentan buenos resultados a la hora de detectar paráfrasis complejas que incluyen sinónimos o expresiones con significados equivalentes. Sin embargo, ofrecen un buen nivel de efectividad teniendo en cuenta la rapidez de procesamiento que presentan.

Las principales métricas basadas en similitud léxica, son las que se basan en el traslape de palabras, es decir, en el número de palabras coincidentes en ambos textos. Las principales son:

- Coeficiente de Dice: este coeficiente calcula la similitud entre dos oraciones dividiendo el número de palabras comunes entre ambas oraciones por el número total de palabras y multiplica el resultado por dos [37].

$$S_{Dice}(O_1, O_2) = 2 \times \frac{|O_1 \cap O_2|}{|O_1| + |O_2|} \quad (2.9.)$$

  Donde O_1 y O_2 son las oraciones que se desean comparar semánticamente, $|O_1 \cap O_2|$ son las palabras comunes a ambas oraciones, $|O_1|$ y $|O_2|$ son el número de palabras de cada oración respectivamente.

- Coeficiente de Jaccard: este coeficiente divide el número de palabras comunes a las dos oraciones por el número de palabras que conforman la unión de los términos de ambas oraciones [38].

$$S_{Jaccard}(O_1, O_2) = \frac{|O_1 \cap O_2|}{|O_1 \cup O_2|} \quad (2.10.)$$

  Donde O_1 y O_2 son las oraciones que se desean comparar semánticamente, $|O_1 \cap O_2|$ son las palabras comunes a ambas oraciones y $|O_1 \cup O_2|$ es el total de palabras que hay entre ambas oraciones.

- Coeficiente del traslape: este coeficiente divide el número de palabras comunes a las dos oraciones por el número de palabras que tenga la oración más corta [39].

$$S_{Traslape}(O_1, O_2) = \frac{|O_1 \cap O_2|}{min(|O_1|, |O_2|)} \quad (2.11.)$$

  Donde O_1 y O_2 son las oraciones que se desean comparar semánticamente, $|O_1 \cap O_2|$ son las palabras comunes a ambas oraciones y $|O_1|$ y $|O_2|$ son el número de palabras de cada oración respectivamente.

- Coeficiente del coseno: este coeficiente divide el número de palabras comunes entre ambas oraciones por la raíz cuadrada del producto del número de palabras de cada oración [40].

$$S_{Coseno}(O_1, O_2) = \frac{|O_1 \cap O_2|}{\sqrt{|O_1||O_2|}} \quad (2.12.)$$

  Donde O_1 y O_2 son las oraciones que se desean comparar semánticamente, $|O_1 \cap O_2|$ son las palabras comunes a ambas oraciones y $|O_1|$ y $|O_2|$ son el número de palabras de cada oración respectivamente.

Técnicas basadas en la similitud semántica de las palabras

Estas técnicas se basan en la comparación de los significados de las palabras de dos oraciones para calcular un valor representativo de su similitud semántica. Para comparar los significados de las palabras de las oraciones, estas técnicas se nutren de grandes volúmenes de información que utilizan para determinar la relación semántica de las distintas palabras.

Estos métodos que utilizan grandes cantidades de información para establecer criterios de comparación semántica entre las palabras, se dividen en dos grandes grupos, según el origen de la información que utilizan: métodos basados en corpus [74] y métodos basados en conocimiento [75].

Los métodos basados en corpus se nutren de grandes colecciones de textos para extraer características de ellos y a partir de éstas calcular los valores de similitud semántica de los textos a evaluar. Una de las herramientas más utilizada que se basa en corpus es el conjunto de datos MSRPC (Microsoft Research Paraphrase Corpus), un corpus formado por 5.801 pares de oraciones extraídas de noticias de Internet. De estos 5.801 pares de oraciones, 1.725 pares están designados como conjunto de prueba y los 4.076 restantes como conjunto de entrenamiento [40]

Todo el conjunto de datos fue clasificado manualmente por expertos que etiquetaban los pares de oraciones según la existencia de paráfrasis entre las oraciones de cada par. El objetivo de los creadores de este corpus era crear un corpus monolingüe de pares de oraciones alineadas [41]. Sin embargo, se han encontrado pruebas que señalan que este corpus no es una herramienta efectiva para detectar paráfrasis si se compara su distribución con la encontrada generalmente en los textos [27].

Los métodos basados en conocimiento utilizan información léxica y semántica proveniente de recursos con conocimiento humano, como diccionarios o tesauros y basándose en este conocimiento establecen su criterio de comparación semántica entre las palabras. La herramienta más utilizada que se basa en conocimiento es WordNet, una base de datos léxica en inglés [42][43] cuyos objetivos son: conformar una combinación de diccionario y

tesauro sencillo e intuitivo y facilitar el análisis textual y el procesamiento del lenguaje natural.

En WordNet, el léxico se organiza en cinco categorías: sustantivos, verbos, adjetivos, adverbios y elementos funcionales. Adicionalmente, las palabras de esta base de datos, están organizadas por Synsets. Los Synsets son conjuntos de palabras que son consideradas sinónimos entre sí y por lo tanto, representan un mismo concepto. No obstante, muchas palabras pueden tener más de un significado y por lo tanto, pertenecer a varios Synsets o pertenecer a más de una categoría gramatical. Para solucionar este hecho, WordNet prioriza las acepciones de una misma palabra y su categoría gramatical teniendo en cuenta la frecuencia con la que se utiliza la palabra en cuestión con cada significado y categoría gramatical [36].

WordNet también ofrece otro tipo de relaciones semánticas entre las palabras como: antonimia, hiperonimia, hiponimia, meronimia y relaciones morfológicas.

Una de las funcionalidades más útiles de WordNet dedicada a la detección de paráfrasis es la capacidad de calcular un valor numérico que represente la similitud entre dos palabras. Esta funcionalidad, en el ámbito de la detección de paráfrasis, es de gran utilidad ya que permite cuantificar la similitud semántica textual entre las palabras de las oraciones que deben ser evaluadas y en base a este valor decidir si existe paráfrasis o no. Esta funcionalidad está implementada por el paquete de WordNet llamado: WordNet.Similarity [36].

WordNet.Similarity ofrece varias métricas que calculan un valor representativo de la similitud semántica entre dos palabras. Estas métricas tienen en cuenta las similitudes entre los significados de las palabras y operan sobre parejas de palabras con la misma categoría gramatical cuyas estructuras sintácticas puedan ser organizadas en jerarquías “es un” como los verbos y los sustantivos. Los adjetivos y los adverbios no están organizados de acuerdo a una jerarquía “es un” por lo que las métricas de similitud no pueden ser aplicadas con ellos [36].

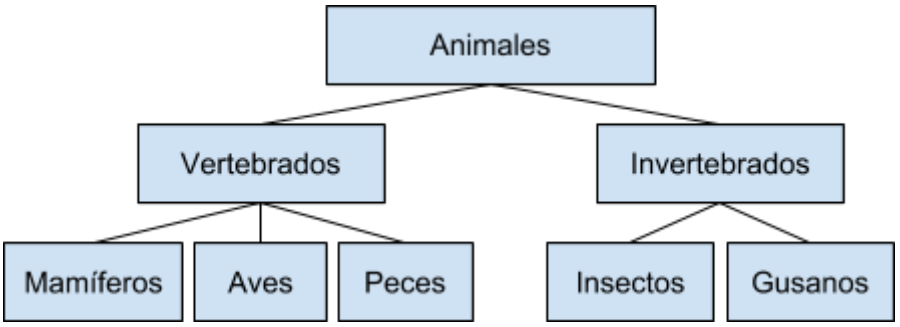

Figura 2.1. Extracto de la Jerarquía “es un” de WordNet.

Además de la jerarquía “es un”, WordNet cuenta con otros métodos para relacionar conceptos, como las relaciones del tipo “parte de” (árbol y bosque) o antítesis (abierto y cerrado) entre otras.

Las métricas que ofrece WordNet.Similarity son nueve: seis que miden la similitud semántica y tres que miden la relación entre las palabras. A continuación se describen las métricas más importantes (una basada en la relación de las palabras y las seis que miden similitud semántica):

- Métrica lesk [44]: esta métrica mide la relación que existe entre dos palabras mediante las relaciones de WordNet existentes en sus definiciones [45]. La relación entre las definiciones es calculada mediante una función que se basa en la búsqueda de la secuencia de palabras más larga que aparece en ambas definiciones o que presentan relaciones de WordNet (antonimia, hiponimia, hiperonimia, etc). Estas secuencias de palabras no pueden comenzar o finalizar con una palabra funcional (preposiciones, pronombres, artículos o conjunciones). El valor que calcula esta función es el cuadrado del número de palabras que tenga la secuencia detectada. Cuando la función detecta una secuencia con una relación de WordNet, almacena la puntuación correspondiente (el cuadrado del número de palabras de la secuencia) y elimina esta secuencia del par de definiciones. Este proceso se repite hasta que no se detectan más secuencias de palabras con relaciones de WordNet.

- Métrica Leacock y Chodorow (lch) [46]: esta métrica calcula la similitud entre dos palabras mediante la distancia entre ellas en una jerarquía “es un” [46]. La función está definida como:

$$S_{lch} = -log(\frac{p}{2d}) \quad (2.13.)$$

  Donde p es la distancia más corta entre las palabras y d es la profundidad máxima de las palabras en la red.

  El resto de métricas se basan en el concepto del ancestro común más específico (LCS) y el contenido informativo (IC) [36]. Dados dos conceptos K_1 y K_2 en una jerarquía “es un”, el LCS es definido como el concepto más específico que ambos tienen en común [47]. El IC de un nodo es una aproximación de cuán informativo es este concepto [48]. El IC está definido como:

$$IC = -log(P(c)) \quad (2.14.)$$

  Donde P(c) es la probabilidad de encontrar a c en un corpus extenso.

- Métrica Wu y Palmer (wup) [47]: esta métrica calcula la similitud entre dos palabras en base a la distancia de los conceptos con el LCS [33][47] y la distancia del LCS con el concepto raíz. La similitud de dos nodos K_1 y K_2 es:

$$S_{wup} = \frac{2 \times D_3}{D_1 + D_2 + 2 \times D_3} \quad (2.15.)$$

Donde D_1 es la distancia de K_1 al LCS, D_2 es la distancia de K_2 al LCS y D_3 la distancia del concepto raíz al LCS.

- Métrica Resnik (res) [48]: esta métrica calcula la similitud entre dos conceptos mediante el IC del LCS de los mismos [48]. La similitud se define como:

$$S_{res} = IC(LCS) \quad (2.16.)$$

- Métrica Lin (lin) [49]: esta métrica proviene de la medida Resnik mediante una normalización y se basa en el IC de los conceptos a comparar [49]. Esta similitud se define como:

$$S_{lin} = \frac{2 \times IC(LCS)}{IC(D_1) \times IC(D_2)} \quad (2.17.)$$

Donde D_1 es la distancia de K_1 al LCS y D_2 es la distancia de K_2 al LCS.

- Métrica Jiang y Conrath (jcn) [50]: esta métrica también se basa en el IC [50]. Esta similitud se define como:

$$S_{jcn} = \frac{1}{IC(D_1) + IC(D_2) + 2 \times IC(LCS)} \quad (2.18.)$$

Donde D_1 es la distancia de K_1 al LCS y D_2 es la distancia de K_2 al LCS.

- Métrica Path length (path) [51]: esta métrica se basa en el número de aristas que contiene el camino más corto entre los dos conceptos a comparar. Esta similitud se define como:

$$S_{path} = SP(IC(D_1), IC(D_2)) \quad (2.19.)$$

Donde SP() es la función que devuelve la cantidad de aristas del camino más corto entre los conceptos requeridos.

A continuación se describen varios métodos implementados que abordan la detección de paráfrasis mediante las técnicas descritas anteriormente relacionadas con la comparación semántica de las palabras.

1. *Método de detección de paráfrasis con métricas de similitud semántica basadas en corpus y conocimiento*

En este trabajo desarrollado por Mihalcea, R., Corley, C. & Strapparava, C. [52] se expone un método que es capaz de calcular la similitud semántica entre textos cortos, por lo que puede ser orientado para decidir si existe paráfrasis entre un par de oraciones. Este método utiliza medidas de similitud semántica entre palabras basadas en corpus y en conocimiento. Mediante estas medidas de similitud semántica entre palabras se pretende tener en cuenta las relaciones de sinonimia, antonimia u otro tipo de relaciones entre los significados de las palabras, además de tener en cuenta las coincidencias léxicas literales a la hora de calcular un valor representativo de la similitud semántica textual entre dos oraciones.

Esta similitud entre dos textos cortos se calcula de la siguiente manera:

$$Sim(T_1, T_2) = \frac{1}{2}(meanT_1 + meanT_2) \quad (2.20.)$$

$$meanT_1 = \frac{\sum_{w \varepsilon (T_1)} (S_{MS}(w, T_2) \times idf(w)}{\sum_{w \varepsilon (T_1)} idf(w)} \quad (2.21.)$$

$$meanT_2 = \frac{\sum_{w \varepsilon (T_2)} S_{MS}(w, T_1) \times idf(w)}{\sum_{w \varepsilon (T_2)} idf(w)} \quad (2.22.)$$

Dónde $S_{MS}(w, T_i)$ retorna el valor máximo de similitud semántica calculado entre la palabra “w” y las palabras del texto corto “Ti” utilizando la métrica de similitud semántica elegida, e idf(w) es la frecuencia inversa de documento [53] de la palabra “w”. La frecuencia inversa de documento [53] se define como:

$$idf(w) = \frac{log(D)}{d} \quad (2.23.)$$

Donde “D” es el número total de documentos en el corpus y “d” es el número total de documentos que incluyen la palabra “w”. En este método el valor de idf(w) se calcula utilizando el Corpus Nacional Británico [54].

En la ecuación 20, que representa el cálculo de similitud semántica se observa que en este método se combinan las métricas de similitud semántica a nivel de palabra de la siguiente manera: en primer lugar, se parte del texto corto T_1 y se compara cada palabra de éste con las palabras del texto corto 2, T_2, con el objetivo de encontrar para cada palabra de T_1 la palabra de T_2 con la que tenga una similitud semántica mayor. En segundo lugar, se repite el mismo proceso pero partiendo de T_2. Después cada similitud semántica seleccionada se pondera con su medida idf correspondiente y se suman. Finalmente se

normaliza con la longitud del texto corto correspondiente y ambas medidas, resultantes de comenzar con las palabras de T_1 y de T_2 se combinan con la media aritmética [52].

Al realizar el cálculo de la similitud semántica entre palabras, se ha reducido la posibilidad de establecer similitud semántica a solo las palabras que pertenecen a la misma categoría gramatical ya que la mayoría de métricas utilizadas en este método no permiten el cálculo de similitud semántica entre palabras con distinta categoría gramatical.

El resultado de este método es un valor comprendido entre 0 y 1 para cada par de oraciones, donde 0 indica que no existe ninguna equivalencia semántica y 1 indica que la similitud semántica es máxima. Para decidir si existe paráfrasis o no, se utiliza un umbral de decisión de 0.5 de manera que, si el valor calculado con la ecuación 2.20. es superior o igual a 0.5 se decide que existe paráfrasis y si es menor se decide que no existe paráfrasis.

En este método se experimenta con varias métricas de similitud semántica a nivel de palabra, tanto basadas en conocimiento como en corpus.

Las métricas basadas en conocimiento son algunas de las métricas implementadas en WordNet:

- Leacock & Chodorow [46]
- Lesk [44]
- Wu & Palmer [47]
- Resnik [48]
- Lin [49]
- Jiang & Conrath [50]

Las métricas basadas en corpus son:

- Información mutua puntual [55]
- Análisis Semántico Latente (LSA) [56]

Para testar el método implementado y calcular los resultados se utilizó el conocido corpus Microsoft Research Paraphrase Corpus (MSRPC) [41]. Los mejores resultados del método descrito se obtuvieron a partir de un cálculo que combinaba varias métricas basadas en conocimiento. Se alcanzó una exactitud de 70,3% y una medida F del 81,3% sobre el corpus MSRPC [52].

2. Método de detección de paráfrasis a partir de la matriz de similitud

Este método presenta un sistema similar al propuesto por Mihalcea, R., Corley, C. & Strapparava, C. [52] para la detección de paráfrasis en pares de oraciones que considera la relación semántica más estrecha para cada palabra de la oración 1 con la oración 2 y viceversa, por lo que se tenía en cuenta una medida de similitud semántica por cada palabra del par de oraciones. Sin embargo, en este otro método, con el objetivo de mejorar

la detección de paráfrasis, se consideran las relaciones semánticas de todas las palabras de la oración 1 con todas las palabras de la oración 2. De esta manera se obtiene una matriz de similitud semántica formada por las similitudes semánticas resultantes de la comparación de las palabras de la oración 1 con las palabras de la oración 2. Posteriormente, en base a esta matriz de similitud semántica se calculará un valor representativo de la similitud semántica entre las dos oraciones a evaluar.

Esta técnica basada en el cálculo y el uso de una matriz de similitud semántica fue propuesta por Stevenson, M. & Greenwood, M. [57] con el objetivo de extraer información. Fernando y Stevenson [31] la utilizan por primera vez con fin de detectar paráfrasis. En este método se utilizaron algunas de las métricas de WordNet:Similarity (Jiang & Conrath [50], Leacock & Chodorow [46], lesk [44], Lin [49], Wu & Palmer [47] y Resnik [48]). Estas métricas de WordNet:Similarity, almacenan varios significados de cada palabra por lo que es imprescindible especificar el significado que debe ser utilizado para cada comparación. En primera instancia, los experimentos se realizaron con el primer significado de cada palabra y posteriormente, se experimentó con todos los significados de cada palabra, pero el tiempo de cálculo de los valores de similitud aumentó notablemente a causa de que muchas palabras cuentan con más de un significado. Además los resultados obtenidos con el uso de todos los significados, no mejoraron suficientemente el rendimiento obtenido con el uso del primer significado por lo que en la implementación del método se utilizó el primer significado de cada palabra.

Los mejores resultados fueron obtenidos con la métrica de WordNet:Similarity: Jiang & Conrath [50]. Obteniendo una exactitud del 74,1% y una medida F del 82,4%.

2.2.2. Métodos basados en aprendizaje automático

Estos métodos enfocan la detección de paráfrasis como un problema de clasificación binaria en el que las posibles soluciones son: paráfrasis o no-paráfrasis. Estos métodos utilizan algoritmos que son capaces de aprender modelos a partir de conjuntos de datos. Estos algoritmos son entrenados con conjuntos de datos debidamente etiquetados de los cuales extraen características y establecen relaciones entre las entradas y las salidas.

El entrenamiento de estos algoritmos es de suma importancia ya que de éste depende la precisión del método. Para que el entrenamiento sea de calidad es necesario contar con un conjunto de datos variado y previamente clasificado.

El algoritmo extraerá características semánticas, sintácticas o léxicas de este conjunto de datos y en función de esta información, podrá clasificar cualquier par de oraciones que no haya sido etiquetado previamente.

A continuación se muestra el esquema general que utilizan los métodos de aprendizaje automático orientados a detectar paráfrasis.

El proceso comienza con un conjunto de pares de oraciones clasificados, normalmente manualmente. Después el algoritmo es entrenado y aprende del conjunto de datos mediante

la extracción de características generando de esta manera un modelo de clasificación. Finalmente, cuando el algoritmo ha generado un modelo de clasificación, es capaz de clasificar un nuevo conjunto de datos.

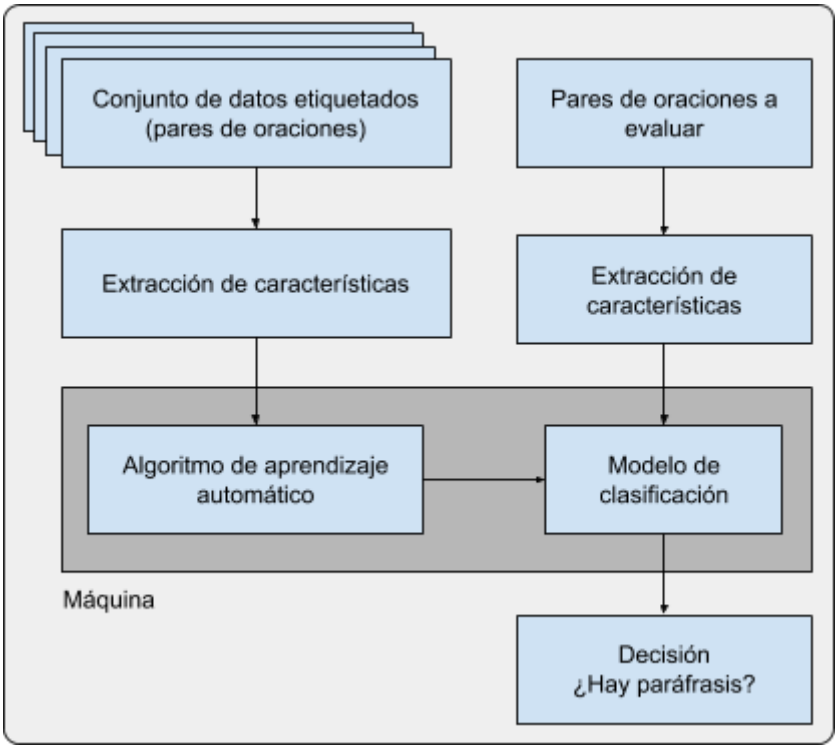

Figura 2.2. Esquema general de los métodos de aprendizaje automático orientados a la detección de paráfrasis.

Este tipo de métodos han resultado ser los más precisos [36] a pesar de la dificultad que supone entrenar los algoritmos con conjuntos de datos completos que ofrezcan todas las clases posibles de paráfrasis.

A continuación se describen varios métodos que abordan la detección de paráfrasis a partir de los principios del aprendizaje supervisado.

1. Método de detección de paráfrasis por disimilitud

Este trabajo fue desarrollado por Qiu, L., Kan, M. & Chua, T. [28] y su objetivo es detectar paráfrasis en pares de oraciones basándose en las diferencias que existen entre éstas. El método consta de dos fases. En primer lugar se dividen las oraciones en unidades con información semántica significativa, llamadas minutas de información. En segundo lugar, se tratan de relacionar las minutas de información de una oración con las de la otra. Si alguna minuta de información queda sin una minuta equivalente en la otra oración, el contenido semántico de ésta es analizado. Y si no queda ninguna minuta sin relacionar, o sus significados son catalogados de irrelevantes, se decidirá que existe paráfrasis.

En este método se utilizan como minutas de información a tuplas predicado-argumento, que consisten en dividir las oraciones separando el verbo predicado como el objetivo y a sus argumentos.

Estas tuplas predicado-argumento se obtienen mediante un analizador sintáctico [58] y el etiquetador de rol semántico ASSERT [59].

El emparejamiento entre las tuplas de ambas oraciones se realiza por comparación directa tomando del objetivo y de los argumentos las palabras claves. En el caso de que queden tuplas sin relacionar se intentan relacionar con heurísticas adicionales como el manejo de la forma copular y sintagmas nominales.

Las tuplas que no sean relacionadas de cada oración, constituyen información diferente respecto a la otra oración del par, por lo que estas deben ser analizadas. Este análisis debe indicar si estas informaciones son relevantes para la paráfrasis o no. Para llevar a cabo este análisis se utiliza aprendizaje automático, en concreto dos tipos del clasificador de máquinas de soporte vectorial: la ruta del árbol de análisis sintáctico y el predicado.

En el momento en el que se desarrolló este método, no existían corpus de entrenamiento para los significados del predicado por lo que se implementó un nuevo método en base a un corpus etiquetado. Se etiquetaron los datos del conjunto de entrenamiento en base a dos criterios: si existe paráfrasis y la procedencia de las tuplas sin emparejar. Estos criterios de clasificación dan lugar a cuatro categorías distintas, de las que solo dos se utilizaron como datos de entrenamiento:

- **Caso 1**: Los casos en los que existe paráfrasis y las tuplas sin emparejar provienen de la misma oración.
- **Caso 2**: Los casos en los que no existe paráfrasis y solamente hay una tupla sin emparejar.

Las tuplas de los pares del caso 1 se etiquetan como insignificantes porque aunque no están emparejadas, existe paráfrasis y las tuplas de los pares del caso 2 se etiquetan como significativas porque esta única tupla por par es la responsable de que no exista paráfrasis.

Este método se probó sobre el corpus MSRPC y presentó buenos resultados, logrando una exactitud del 72,9% y una medida F del 81,6%.

2. *Método de detección de paráfrasis por canonicalización de textos*

Este trabajo fue desarrollado por Zhang, Y. & Patrick, J. [27]. La tesis central de este método es la detección de paráfrasis mediante la transformación de las oraciones a evaluar a sus formas canónicas (transformación de las palabras a su raíz). Esta idea se basa en que las oraciones con significados equivalentes tienen más probabilidad de transformarse en las mismas oraciones canónicas que oraciones con significados distintos.

Para llevar a cabo este método, se utilizaron tres tipos de técnicas de canonicalización: la sustitución de los números por etiquetas genéricas, la conversión de voz pasiva en activa y la sustitución de los tiempos futuros por la palabra “will”.

Cuando se ha transformado el texto a su forma canónica, se procede a crear un modelo de aprendizaje automático supervisado basado en un árbol de decisión que decide si existe paráfrasis o no en cada par de oraciones. Para crear este modelo son utilizadas características léxicas como la subsecuencia común más larga, la distancia de edición y una medida basada en n-gramas [27].

Este trabajo obtuvo el mejor resultado utilizando la conversión de voz pasiva en activa, logrando una exactitud del 71,9% y una medida F del 80,7% probado sobre el conjunto de pruebas de MSRPC [27].

3. Método de detección de paráfrasis combinando medidas de similitud

Este trabajo fue implementado por Malakasiotis, P. [32] y consiste en tres métodos enfocados en la detección de paráfrasis. El trabajo se basa en un clasificador automático de máxima entropía para aprender cómo combinar varias medidas de similitud entre oraciones. A continuación se describen los tres métodos [32]:

- **INIT**: este método consta de nueve medidas de similitud: distancia de Levenshtein, distancia de Jaro-Winkler, distancia de Manhattan, distancia Euclidiana, similitud del coseno, distancia de n-gramas (con n = 3), coeficiente de Dice, Coeficiente de Jaccard y coeficiente de coincidencia.

- **INIT + WN**: este método, mediante WordNet, detecta los sinónimos y los procesa como palabras equivalentes, por lo que supone una mejora de INIT.

- **INIT + WN + DEP**: la diferencia de este método con los anteriores es que en este se tienen en cuenta las relaciones gramaticales. Estas relaciones gramaticales se añaden a través de tres medidas que son capaces de cuantificar estas relaciones de similitud. Estas medidas son: recuerdo de dependencia de S_1 (R_1), recuerdo de dependencia de S_2 (R_2) y su medida F (F_{R_1, R_2}), definidas como:

$$R_1 = \frac{|dependencias\ comunes|}{|dependencias\ de\ S_1|} \quad (2.24.)$$

$$R_2 = \frac{|dependencias\ comunes|}{|dependencias\ de\ S_2|} \quad (2.25.)$$

$$F_{R_1, R_2} = \frac{2 \times R_1 \times R_2}{R_1 + R_2} \quad (2.26.)$$

El método INIT + WN + DEP, al ser el más completo obtuvo los resultados más precisos, logrando una exactitud del 76,1% y una medida F del 82,8% [32].

4. *Método de detección de paráfrasis mediante métricas de traducción automática*

Este trabajo fue desarrollado por Madnani, N., Tetreault, J. & Chodorow, M. [13] y en él se utilizan métricas desarrolladas originalmente para la traducción automática con el objetivo de detectar paráfrasis desde un enfoque supervisado. El método utiliza los clasificadores de regresión logística, k-vecinos más cercanos y máquinas de soporte vectorial. Las técnicas de evaluación de traducción automática utilizadas son las siguientes:

- **BLEU** [60]: esta métrica se basa en el traslape de n-gramas, con distintos valores de "n".
- **NIST** [61]: también funciona con n-gramas pero combina un promedio aritmético y uno geométrico de los n-gramas compartidos entre el total.
- **TER** [62]: distancia de edición que calcula el mínimo número de operaciones para transformar una traducción a evaluar en la traducción ideal. Las operaciones son sustituir, insertar y eliminar.
- **TERp** [63]: métrica extendida de TER que añade operaciones de stemming, sinonimia y paráfrasis.
- **METEOR** [64]: métrica basada en n-gramas que tiene en cuenta tanto la precisión como el recuerdo. Además realiza un pre-procesado en el que añade operaciones de stemming, sinonimia y paráfrasis.
- **SEPIA** [65]: utiliza n-gramas estructurales capaces de recopilar más información que los n-gramas tradicionales.
- **MAXSIM** [66]: esta métrica se basa en el alineamiento de las palabras de ambas oraciones, emparejando cada palabra de una oración con la palabra más similar a ésta de la otra oración.

Individualmente la métrica TERp resultó ser la que obtuvo los mejores resultados, con una exactitud del 74,3% mejorando varios métodos basados en la similitud de las palabras [31][67][52] y algunos que se basan en aprendizaje automático supervisado [27][28]. Llevando a cabo la técnica de combinación de todas estas métricas se obtuvo un resultado que supone uno de los mejores en la detección de paráfrasis [13], con una exactitud del 77,4% y una medida F del 84,1%.

2.3. Conclusiones

En este capítulo se han recopilado las principales herramientas, técnicas y métodos orientados a la detección de paráfrasis. Para llevar a cabo esta revisión del estado del arte de la detección de paráfrasis se ha diferenciado entre dos vertientes: los métodos que utilizan directamente la similitud entre las palabras para calcular valores representativos de la similitud semántica de un par de oraciones y en base a estos valores, decidir si existe

paráfrasis; y los métodos basados en aprendizaje automático supervisado que son capaces de extraer características de diversas naturalezas de un conjunto de datos y aprenderlas con el objetivo de clasificar cualquier par de oraciones en base a estos criterios aprendidos.

En la solución propuesta en el Presente Trabajo de Fin de Grado se ha implementado un método con las características de la primera vertiente: la similitud entre las palabras de las oraciones entre las que se debe decidir la existencia de paráfrasis. Para esto se han combinado varias técnicas y algunas han sido utilizadas en los métodos descritos anteriormente.

El trabajo desarrollado por Mihalcea, R., Corley, C. & Strapparava, C. [52] está estrechamente relacionado con la solución propuesta en el presente Trabajo de Fin de Grado ya que en ella también utilizó las métricas de similitud semántica textual de WordNet para calcular un valor representativo de la similitud semántica textual que influye en el cálculo del valor final a partir del cual se decide si existe paráfrasis.

Sin embargo, la parte relacionada con las métricas de similitud semántica de la solución propuesta en el Presente Trabajo de Fin de Grado, no incluye las métricas de similitud semántica basadas en corpus que se incluyen en el trabajo desarrollado por Mihalcea, R., Corley, C. & Strapparava, C. [52] (Información mutua puntual y Análisis semántico latente) que podrían mejorar los resultados obtenidos con la solución propuesta en el Presente Trabajo de Fin de Grado. No obstante, la solución presentada en el Presente Trabajo de Fin de Grado incluye Word2Vec, una herramienta que a partir de un corpus de información es capaz de crear un modelo de tópicos con el que calcular valores representativos de la similitud semántica entre parejas de palabras. Con un corpus completo, es posible generar un modelo de tópicos con prestaciones comparables a las métricas de similitud semántica basadas en corpus como Información mutua puntual y Análisis semántico latente.

El trabajo de Fernando y Stevenson [31] también está relacionado con la solución propuesta en el presente Trabajo de Fin de Grado ya que en ella también ha sido implementada la idea de la matriz de similitud para calcular un valor representativo de la similitud semántica textual entre un par de preguntas que, aunque no supone el valor final con el que se decide la existencia de paráfrasis, influye directamente en su cálculo.

Las diferencias de este método aquí descrito y la solución propuesta en el presente Trabajo de Fin de Grado son el uso de Word2Vec para calcular los valores de similitud que componen la matriz de similitud con la que se calculan los valores definitivos que representan la similitud entre los pares de preguntas. No obstante, las métricas Wu & Palmer [47] y Path Length [51] basadas en WordNet también han sido utilizadas para calcular los valores que conforman la matriz de similitud semántica. De esta manera, la solución propuesta en el presente Trabajo de Fin de Grado combina técnicas de varias naturalezas para calcular los valores que conforman la matriz de similitud.

Los métodos explicados anteriormente que se basan en el aprendizaje automático supervisado para detectar paráfrasis, han reportado unos resultados más precisos que el resto de métodos. Estos métodos basados en el aprendizaje automático supervisado

ofrecen más oportunidades para investigar y experimentar con las diferentes características que pueden ser extraídas de los textos.

Además, se prevé que en los próximos años conforme los métodos de extracción de características y los corpus de entrenamiento que utilizan los métodos basados en aprendizaje automático supervisado evolucionen y se vuelvan más precisos y eficientes estos métodos que utilizan estas técnicas de clasificación tenderán a la comprensión y a criterios practicamente humanos.

Sin embargo, en la actualidad, los métodos que se basan en el aprendizaje automático supervisado pueden ser complementados con técnicas de similitud entre palabras de una manera efectiva ya que la información recopilada por las distintas técnicas enriquece la solución final.

Por tanto, pese a que el aprendizaje automático supervisado supone un gran avance en cuanto a la detección de paráfrasis y se le prevé aún un considerable margen de mejora, en este momento, la combinación de técnicas basadas en estas ideas con las técnicas implementadas en la solución propuesta en el presente Trabajo de Fin de Grado nos llevarán a la implementación de métodos más precisos y fiables.

3. Problema planteado y solución propuesta

3.1. Problema planteado

En el presente trabajo se plantea un problema de detección de paráfrasis en parejas de preguntas. Por lo tanto, la solución del problema consiste en identificar los pares de preguntas cuyos enunciados sean equivalentes semánticamente (ambas preguntas significan lo mismo) y los pares de preguntas que no lo sean.

Este problema es equivalente a uno propuesto en uno de los retos de Kaggle (www.kaggle.com) una plataforma web que organiza competiciones relacionadas con "machine learning". Para el desarrollo de la solución del problema, en este reto se proporcionaban dos colecciones de datos: una de entrenamiento que consistía en 404.290 pares de preguntas y otra de pruebas que consistía en 1.048.575 pares de preguntas, todas ellas en inglés.

El conjunto de datos de entrenamiento estaba debidamente etiquetado, es decir, los pares cuyas preguntas eran equivalentes semánticamente y los pares cuyas preguntas no lo eran estaban identificados, ya que esta colección de datos estaba diseñada para entrenar la solución propuesta. Por el contrario, el conjunto de datos de pruebas no estaba etiquetado, los pares de preguntas no estaban identificados ya que esta colección de datos estaba diseñada para testar la solución final y dado que este problema formaba parte de una competición, solo Kaggle tenía esta información.

Una vez finalizado el plazo de entrega de las soluciones al reto de Kaggle, la colección de datos de pruebas, que no estaba etiquetada según la equivalencia semántica de sus preguntas, no tenía ninguna utilidad para el desarrollo de la solución que se propone en el presente Trabajo de Fin de Grado ya que no permitía comprobar la precisión de los métodos implementados.

Dada la necesidad de tener los pares de preguntas etiquetados según la equivalencia semántica de sus preguntas para poder comprobar la precisión de los métodos implementados y así continuar mejorando la solución (entrenando), la colección de datos de entrenamiento ha sido el único conjunto de datos utilizado para el desarrollo de la solución aquí propuesta.

Para garantizar la calidad de la solución y evitar el sobreentrenamiento, se llevó a cabo una partición de los datos del conjunto de entrenamiento. Los primeros 270.000 pares de preguntas han sido utilizados como datos de entrenamiento para desarrollar los métodos que componen la solución y los 134.290 siguientes han sido utilizados como datos de prueba para obtener los porcentajes de fiabilidad de cada método de la solución y de la solución final durante los experimentos llevados a cabo.

3.2. Enfoque de la solución

Con el objetivo de desarrollar una solución efectiva para el problema propuesto se ha llevado a cabo la partición de datos descrita en la sección anterior. De esta manera los métodos que componen la solución serán diseñados y probados con los datos de entrenamiento mientras que con los datos de prueba se calculará la fiabilidad y la precisión de estos métodos.

La solución del problema consta de varias fases. En primer lugar se llevará a cabo un pre-procesado de los datos disponibles, en segundo lugar, una vez los datos están pre-procesados, se procederá a evaluar los datos para calcular un valor para cada par de preguntas que representará la similitud semántica entre las mismas. Finalmente tendrá lugar la decisión, que consiste en decidir en base al valor calculado en la fase de evaluación qué pares de preguntas contienen preguntas equivalentes semánticamente y qué pares no. Para llevar a cabo esta decisión de forma efectiva, en base al valor calculado durante la fase de evaluación, es imprescindible el cálculo del umbral óptimo de decisión.

Un umbral de decisión es un valor numérico que nos permitirá diferenciar entre los pares de preguntas que clasificaremos como equivalentes semánticamente y los que no. De forma que los pares de preguntas que presenten un valor superior al umbral calculado, serán clasificados como equivalentes (“1”) y los que presenten un valor inferior al umbral serán clasificados como no equivalentes (“0”).

El cálculo de este umbral consiste en un proceso iterativo en el que se dan valores al umbral de decisión. Para comenzar se da un valor al umbral de decisión y se comienzan a clasificar los pares de preguntas en función de este umbral.

Una vez se hayan clasificado todos los pares de preguntas como equivalentes (“1”) o no equivalentes (“0”), se procederá a calcular el porcentaje de acierto de la decisión llevada a cabo. Este porcentaje se calculará comparando la clasificación realizada (a partir de los valores de similitud semántica calculados durante la fase de evaluación) con la clasificación correcta que viene dada en la colección de datos de entrenamiento gracias al etiquetado.

Este porcentaje medirá la cantidad de pares de preguntas clasificados correctamente, ya sea como equivalente (“1”) o como no equivalente (“0”), es decir, medirá la exactitud de la decisión realizada.

Este proceso se repetirá variando el valor del umbral de decisión de forma que obtendremos tantos porcentajes de acierto como valores demos al umbral. El umbral de decisión que conduzca al mayor porcentaje de acierto, será el umbral óptimo para estos valores de similitud semántica, ya que este umbral es el que nos lleva a lograr la mejor clasificación posible de los pares de preguntas, a partir de los valores de similitud semántica calculados en la fase de evaluación.

Una vez calculado el umbral y diseñados los métodos de pre-procesado y de evaluación se puede proceder a probar la fiabilidad y precisión de la solución implementada.

Esta prueba final de la solución debe ser llevada a cabo con el conjunto de datos de prueba con el objetivo de que la prueba sea lo más fiable posible y que la solución no esté sobreentrenada. Por lo tanto se probará la solución implementada con el conjunto de datos de prueba y utilizando el umbral, que ha sido calculado con el conjunto de datos de entrenamiento, para clasificar los valores finales que sean obtenidos durante la fase de evaluación. De esta manera se obtendrá la decisión final para cada par de preguntas.

En este caso, el conjunto de datos de prueba utilizado es un subconjunto del conjunto de datos de entrenamiento del reto de Kaggle, por lo tanto, también está debidamente etiquetado y podremos calcular el porcentaje de acierto de la solución propuesta al problema planteado.

A continuación se incluye un diagrama de bloques que ilustra las diferentes fases en las que está dividida la solución del problema planteado:

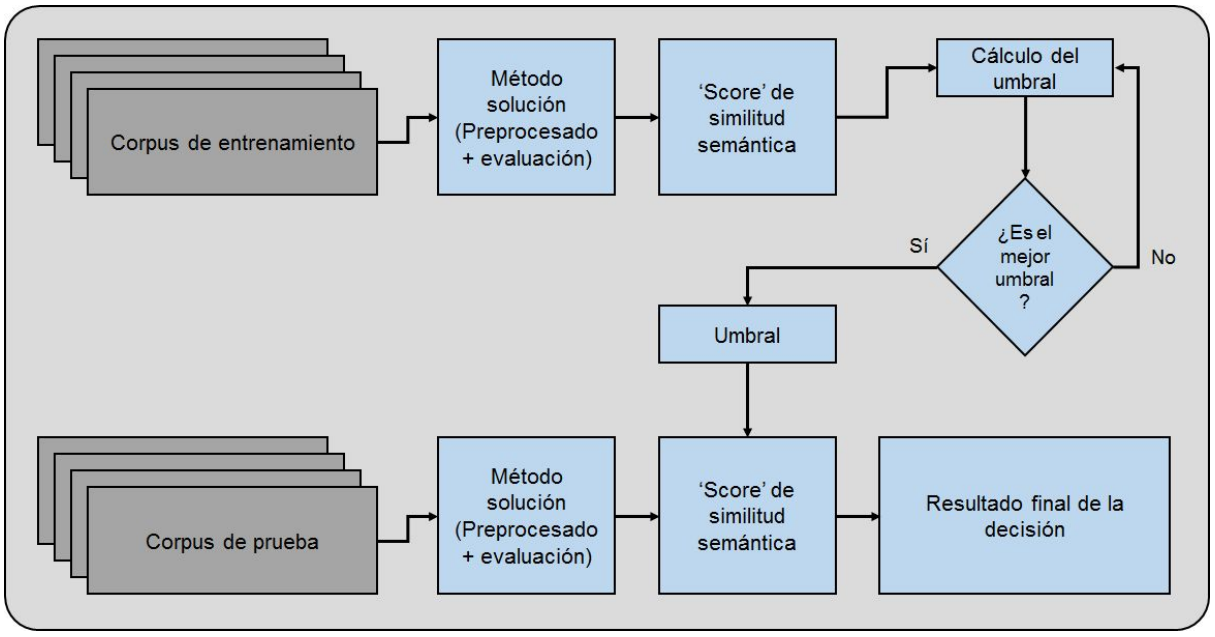

Figura 3.1. Diagrama de bloques de las fases de la solución.

3.2.1. Pre-procesado

El pre-procesado de los datos es la primera etapa de la solución del problema propuesto. Antes de realizar alguna operación con los datos del problema, es muy importante que estos tengan la forma adecuada para llevar a cabo fácil y correctamente las operaciones pertinentes. La forma adecuada depende de las tareas que se vayan a realizar posteriormente con los datos. Por lo tanto, el pre-procesado depende directamente de las tareas que se llevarán a cabo a continuación.

En este caso, los métodos propuestos en la fase de evaluación para calcular los valores de similitud textual entre las preguntas de cada par solamente operan con palabras por lo que

en primer lugar, las frases se segmentan en tokens. Los tokens son equivalentes a palabras, signos de puntuación, números u otros caracteres separados por espacios. En segundo lugar, para facilitar la comparación de palabras, todas las letras se convierten a letras minúsculas y son eliminados los tokens que se corresponden con signos de puntuación u otros caracteres que no nos aportan información semántica. También son eliminadas las denominadas 'stop words': palabras que no aportan información semántica como artículos, pronombres, preposiciones, conjunciones, etc. Por último, los 'tokens' restantes, son sometidos a un proceso de lematización lo que significa que las palabras que se encuentren en su forma flexionada serán reducidas a su lema (raíz). De esta forma se aumentará la posibilidad de coincidencia entre las palabras.

3.2.2. Evaluación

La Evaluación de los datos es la segunda etapa de la solución del problema propuesto. Una vez los datos han sido tratados y presentan el formato adecuado, se procede con la evaluación. La etapa de evaluación consiste en realizar las operaciones necesarias con nuestros datos con el fin de obtener un indicador numérico ('score') que nos aporte la información necesaria acerca de la similitud semántica textual de cada par de preguntas.

En el caso del problema planteado se realizarán las operaciones necesarias sobre las palabras de cada frase con la finalidad de obtener un indicador numérico por cada par de preguntas. Estos indicadores nos deberán aportar información referente a la similitud semántica de cada par de preguntas, de tal manera que estos sean proporcionales a la similitud semántica de las preguntas de cada par. Todos los indicadores estarán normalizados, es decir, su valor estará comprendido entre 0 y 1.

Para alcanzar una solución precisa y fiable al problema planteado, se han implementado varios métodos distintos, basados en distintas técnicas para calcular similitud semántica textual. A partir de todos estos métodos que devolverán un valor de similitud semántica textual correspondiente a cada par de preguntas, se calculará un valor único para cada par de preguntas que será el valor final ("score") con el que junto al umbral calculado, decidiremos qué pares de preguntas clasificaremos como equivalentes "1" y que pares de preguntas clasificaremos como "no equivalentes".

Para calcular un solo valor final (S_F) que represente la similitud semántica textual de cada par de preguntas se computará la media aritmética entre los valores calculados en los distintos métodos implementados (S_{M_1}, S_{M_2}, S_{M_3} y S_{M_4}):

$$S_F = \frac{S_{M_1} + S_{M_2} + S_{M_3} + S_{M_4}}{4} \quad (3.1)$$

Además de estos cuatro métodos, se presenta un quinto método (Método híbrido) en el que se combinan algunas de las técnicas utilizadas en los demás métodos y se añade una métrica que no ha sido implementada en los métodos anteriores con el fin de comparar el

rendimiento de un método que combina varias técnicas con varios métodos combinados mediante la media aritmética. La solución que ofrezca un resultado más preciso será elegida como la solución final del problema planteado.

A continuación se explicarán todos los métodos implementados que han sido utilizados para el cálculo de la solución propuesta del problema planteado.

Método 1. Comparador literal de palabras.

Este método se basa en la repetición de palabras (traslape de palabras) que aparecen en las dos oraciones de un mismo par para calcular un valor que represente la similitud semántica textual entre ambas preguntas.

Dentro de este método pueden ser calculados cuatro valores de la similitud semántica textual para cada par de preguntas según la métrica de cálculo que se aplique. Estas cuatro métricas que calculan la similitud semántica textual basándose en el traslape de palabras son: Coeficiente de Dice [37], Coeficiente de Jaccard [38], Coeficiente de traslape [39] y Coeficiente del Coseno [40].

Los valores que calculan estos coeficientes ya están normalizados con un factor de normalización propio de cada uno, de manera que los valores obtenidos por cada coeficiente para cada par de preguntas se encuentran entre 0 y 1.

Para calcular un único valor representativo de la similitud semántica por cada par de preguntas y no cuatro valores diferentes, se lleva a cabo la media aritmética de estos cuatro valores. El Score final traslape (S_{FT}) será el valor representativo de la similitud semántica de cada par de preguntas y se calculará de la siguiente manera:

$$S_{FT} = \frac{S_{DICE} + S_{JACCARD} + S_{TRASLAPE} + S_{COSENO}}{4} \quad (3.2.)$$

Método 2. Método de conteo

Este método también se basa en la coincidencia literal de palabras que existe en ambas preguntas de cada par, para calcular un valor que represente la similitud semántica textual y mediante el cual decidir si hay paráfrasis. Este método ofrece una manera de calcular llamada “método de conteo” y está inspirado en el método propuesto en Inteligencia Analítica [68].

Este método requiere un pre-procesado previo para facilitar el cálculo necesario. En primer lugar, se debe crear un vector (vector de palabras: V_P) que contenga todas las palabras contenidas en el par de preguntas a evaluar, sin incluir palabras repetidas. A continuación, se deben crear dos vectores más (V_{P_1} y V_{P_2}), cada uno correspondiente a cada una de las preguntas del par a evaluar. Estos vectores correspondientes a las preguntas del par a

evaluar tendrán la misma longitud que el primer vector creado que contiene todas las palabras contenidas en el par de preguntas y cada posición de estos corresponderá con una de las palabras del vector de palabras. En estos vectores (V_{P_1} y V_{P_2}) se almacenará el número de veces que se repite cada palabra del vector de palabras en la pregunta 1 y en la pregunta 2 respectivamente.

Finalmente, hay que crear un último vector (vector resultado: V_R) que será el vector donde se almacenará la información con la que se calculará el valor representativo de la similitud semántica textual del par de preguntas. Este vector resultado tendrá la misma longitud que los vectores anteriores y en cada posición de este se almacenará el producto de los valores correspondientes de V_{P_1} y V_{P_2}.

Una vez completado el vector resultado, para obtener el valor representativo de la similitud semántica del par de preguntas correspondiente, se sumarán todos los valores que constituyen este vector resultado y posteriormente se procederá a normalizar este valor para que esté comprendido entre 0 y 1. El método de normalización consiste en dividir este valor calculado previamente por el producto de la longitud del vector de palabras y la longitud de la pregunta con menos palabras del par de preguntas a evaluar. Una vez aplicados estos cálculos se obtendrá el valor final representativo de la similitud semántica entre ambas preguntas de un mismo par (S_{FC}).

A continuación se incluye un ejemplo del método de conteo en el que se podrá visualizar todo el proceso de éste:

Par de preguntas a evaluar:

1. “How can I increase the speed of my internet connection while using a VPN?”
2. “How can Internet speed be increased by hacking through DNS?”

Una vez aplicado el pre-procesado implementado en la solución propuesta, el par de preguntas presentado para este ejemplo presenta la siguiente forma:

1. ['increas', 'speed', 'internet', 'connect', 'use', 'vpn']
2. ['internet', 'speed', 'increas', 'hack', 'dns']

V_P	V_{P_1}	V_{P_2}	V_R
‘increas’	1	1	1
‘speed’	1	1	1
‘internet’	1	1	1
‘connect’	1	0	0
‘use’	1	0	0
‘vpn’	1	0	0
‘hack’	0	1	0
‘dns’	0	1	0
RESULTADOS	6	5	**3**

Tabla 3.1. Ejemplo Método de conteo.

$$S_{FC} = \frac{V_R}{(|V_P| \times min(|Pregunta\ 1|, |Pregunta\ 2|)} = \frac{3}{8 \times 5} = 0,075$$

Método 3. Comparador semántico de palabras.

Este método se basa en la comparación semántica de las palabras de las preguntas de cada par para calcular un valor numérico que represente la similitud semántica textual entre ambas preguntas.

Para calcular este valor numérico se comparan todas las palabras de una pregunta con todas las palabras de la otra pregunta del mismo par, es decir, se calcula la matriz de similitud semántica entre las palabras de ambas frases. Este método de cálculo del valor representativo de la similitud semántica está inspirado en el trabajo de Fernando y Stevenson [31] en el que se utilizó por primera vez la matriz de similitud semántica con el objetivo de detectar paráfrasis.

Posteriormente este valor debe ser normalizado. Para esto, es necesario calcular otros dos valores de similitud semántica, uno con cada pregunta del par de preguntas a evaluar.

Para calcular estos valores de normalización se utilizarán unos valores calculados de manera similar a los valores principales descritos anteriormente. Estos valores de normalización son dos y cada uno corresponde a una de las preguntas del par. El cálculo de estos valores se llevará a cabo con la comparación de todas las palabras de una pregunta con las palabras de ella misma. Es decir, se compararán todas las palabras de la pregunta 1

con todas las palabras de la pregunta 1 y el mismo proceso con la pregunta 2, de modo que se obtendrán dos valores de similitud correspondientes a la pregunta 1 y a la pregunta 2 de cada par.

Una vez han sido calculados estos tres valores: el valor principal combinando las palabras de ambas preguntas y los dos valores de normalización, se procederá al cálculo del valor normalizado que representará la similitud semántica de cada par de preguntas.

Para obtener este valor (S_{SN}) hay que dividir el valor principal calculado con las palabras de ambas preguntas por la raíz cuadrada del producto de los otros dos valores calculados con las palabras de la pregunta 1 y la pregunta 2 respectivamente.

$$S_{SN}(p_1, p_2) = \frac{S_{SEM}(p_1, p_2)}{\sqrt{S_{SEM}(p_1, p_1) \times S_{SEM}(p_2, p_2)}} \quad (3.3.)$$

Donde $S_{SEM}(p_1, p_2)$ es la función que calcula la matriz de similitud entre p_1 y p_2.

Para llevar a cabo este cálculo es necesario contar con al menos una métrica de similitud semántica que, dada una pareja de palabras devuelva un valor que represente su similitud semántica. En este caso, han sido utilizadas cuatro métricas distintas, tres de ellas pertenecientes a la herramienta WordNet (Jiang & Conrath [50], Path length [51] y Wu & Palmer [47]) y una de ellas mediante la herramienta Word2Vec. Ambas tecnologías serán explicadas a continuación en el apartado 3.3. Tecnologías utilizadas.

De esta manera, se obtendrán cuatro valores diferentes (S_{jcn}, S_{wup}, S_{path}, S_{W2V}) correspondientes a las cuatro métricas utilizadas para calcular similitud semántica entre un par de palabras. Con el objetivo de obtener un solo valor de similitud semántica textual (S_{FS}) por cada par de preguntas se lleva a cabo la media aritmética de los cuatro valores calculados:

$$S_{FS} = \frac{S_{jcn} + S_{wup} + S_{path} + S_{W2V}}{4} \quad (3.4.)$$

Método 4. Máxima similitud

Este método, al igual que el método 3, se basa en la similitud semántica de las palabras para calcular un valor representativo de la similitud semántica entre las preguntas de cada par a evaluar. Este método está inspirado en el trabajo de Mihalcea, R., Corley, C. & Strapparava, C. [52] en el que se propone un método enfocado al cálculo de similitud semántica entre textos cortos similar al expuesto en este método.

En este método, para calcular el valor representativo de la similitud semántica entre las preguntas de cada par, se busca el mayor valor de similitud semántica para cada palabra de las preguntas del par, con las palabras de la otra pregunta. Es decir, se comienza con la

pregunta 1 y palabra por palabra se busca la palabra más parecida semánticamente en la pregunta 2, hasta tener tantos valores de similitud semántica como palabras tiene la pregunta 1. Posteriormente se repite el mismo proceso pero partiendo de las palabras de la pregunta 2, obteniendo en total un valor de similitud semántica para cada palabra del par de preguntas.

Sumando todos estos valores obtendremos un valor representativo de la similitud semántica del par de preguntas $(S_{ms}(p_1, p_2))$, sin embargo este valor debe ser normalizado para que pueda ser comparable con los valores del resto de pares de preguntas.

Para normalizar este valor obtenido se dividirá por el número de palabras que contengan el par de preguntas.

$$S_{msN}(p_1, p_2) = \frac{S_{ms}(p_1, p_2)}{|p_1| + |p_2|} \quad (3.5.)$$

Donde $p_1 y\ p_2$ son las preguntas del par a evaluar, $S_{ms}(p_1, p_2)$ devuelve el valor de similitud semántica sin normalizar mediante el cálculo descrito anteriormente y $|p_1|$ y $|p_2|$ son el número de palabras que tienen la pregunta 1 y la pregunta 2 respectivamente.

Para llevar a cabo este cálculo, al igual que en el método 3, es necesario contar con al menos una métrica de similitud semántica que, dada una pareja de palabras devuelva un valor que represente su similitud semántica.

En este caso se han utilizado las mismas métricas que en el método 3: tres pertenecientes a la herramienta WordNet (Path Length [51], Wu & Palmer [47] y Jiang & Conrath [50]) y una de ellas mediante la herramienta Word2Vec. Ambas tecnologías serán explicadas a continuación en el apartado 3.3. Tecnologías utilizadas.

Como se ha explicado anteriormente en el método 3, se obtendrán cuatro valores diferentes ($S_{msNpath}$, S_{msNjcn}, S_{msNwup}, S_{msNW2V}) para cada par de preguntas, correspondientes a las cuatro métricas utilizadas (tres correspondientes a WordNet [47][50][51] y una a Word2Vec) por lo que será necesario un cálculo que combine los cuatro valores para obtener un solo valor que represente la similitud semántica textual entre las preguntas de cada par. Para combinar estos cuatro valores se computará la media aritmética entre ellos, obteniendo así un solo valor de similitud semántica por cada par de preguntas S_{FMS}

$$S_{FMS} = \frac{S_{msNpath} + S_{msNjcn} + S_{msNwup} + S_{msNW2V}}{4} \quad (3.6.)$$

Método 5. Método híbrido

Este método combina técnicas basadas en la similitud semántica y en la similitud léxica para calcular un valor representativo de la similitud semántica textual entre las preguntas de

cada par. Además incluye en el cálculo de este valor representativo de la similitud semántica textual una métrica de disimilitud semántica con el fin de acentuar la diferencia cuantitativa en el valor calculado entre los pares en los que las preguntas son equivalentes semánticamente y en los que no.

En primer lugar, este método calcula un valor basado en la similitud léxica de las palabras (S_{SL}), concretamente, se basa en la repetición de palabras entre ambas preguntas de cada par. Para calcular este valor, se implementa el método 1 Comparador literal de palabras que como se ha explicado anteriormente, combina cuatro coeficientes distintos y efectúa la media aritmética para obtener el valor final que representa la similitud textual.

En segundo lugar se calcula un valor basado en la similitud semántica de las palabras. Para llevar a cabo el cálculo de este valor se utilizan solamente las palabras que no aparecen en ambas preguntas, es decir, antes de calcular este valor, se eliminan de los vectores correspondientes a cada pregunta de un mismo par, las palabras que aparecen en ambos vectores. Esta operación previa al cálculo se lleva a cabo para no incluir dos veces la información que nos transmiten las palabras repetidas en ambas preguntas, ya que esta información se almacena en el valor calculado mediante técnicas basadas en la similitud léxica de las palabras que fue explicado anteriormente.

Una vez tenemos a nuestra disposición las palabras de cada pregunta de un mismo par que no coinciden con las de la otra pregunta, se procede a calcular el valor que represente la similitud semántica. Para calcular este valor se implementa la misma idea en la que se basa el Método 4. Máxima similitud. Es decir, se comparan las palabras no coincidentes de la pregunta 1 con las palabras no coincidentes de la pregunta 2 y se almacena solamente el mayor valor de similitud semántica textual por cada palabra de la pregunta 1 y a continuación se lleva a cabo el mismo proceso pero partiendo de las palabras no coincidentes de la pregunta 2. Esta parte del método está inspirada en el trabajo de Mihalcea, R., Corley, C. & Strapparava, C. [52] en el que se propone un método enfocado al cálculo de similitud semántica entre textos cortos similar al expuesto en esta parte del método.

Sin embargo en este caso, se ha introducido una diferencia respecto al método de máxima similitud. En el caso de que la máxima similitud de una de las palabras sea inferior a 0.5 este valor será restado al valor total de similitud de ese par de preguntas en vez de ser sumado. Con esta métrica, se almacenará la disimilitud semántica entre las palabras, ya que se considera que si el valor de similitud semántica calculado es inferior a 0.5, las palabras son lo suficientemente diferentes como para que esta información sea relevante a la hora de decidir la existencia de paráfrasis en los pares de preguntas. Además, en base a los experimentos realizados se ha concluido que esta información que representa la disimilitud semántica entre las palabras es de utilidad en la decisión de la existencia de paráfrasis.

Después de ejecutar este proceso tendremos dos valores de similitud semántica textual correspondientes a cada par de preguntas, calculados en base a técnicas de comparación semántica de las palabras: uno partiendo de las palabras de la pregunta 1 y otro partiendo

de las palabras de la pregunta 2. Con estos dos valores (S_{SS_1} y S_{SS_2}) se efectuará la media aritmética con el fin de obtener un solo valor (S_{SS}) para cada par que represente la similitud semántica textual entre las preguntas de este.

$$S_{SS} = \frac{S_{SS_1} + S_{SS_2}}{2} \quad (3.7.)$$

Finalmente, con este valor resultante de la media aritmética entre los dos valores obtenidos con técnicas de similitud semántica y con el valor calculado con técnicas de comparación literal de palabras se vuelve a efectuar la media aritmética para obtener un solo valor (S_{FMH}) que represente la similitud semántica textual de cada par de preguntas.

$$S_{FMH} = \frac{S_{SS} + S_{SL}}{2} \quad (3.8.)$$

Para llevar a cabo el cálculo de los valores basados en técnicas que utilizan la similitud semántica entre las palabras, al igual que en los métodos anteriores que utilizan este tipo de técnicas, es necesario contar con métricas de similitud semántica que cuantifiquen el valor de la misma. En este caso, al igual que en los métodos anteriores han sido utilizadas cuatro métricas diferentes, tres de ellas pertenecientes a la herramienta WordNet (Path Length [51], Wu & Palmer [47] y Jiang & Conrath [50]) y una de ellas mediante la herramienta Word2Vec. Ambas tecnologías serán explicadas a continuación en el apartado 3.3. de Tecnologías utilizadas.

3.2.3. Decisión

La decisión es la tercera y última etapa que compone la solución propuesta al problema planteado. Una vez tenemos los valores finales (normalizados entre 0 y 1), que representan la similitud semántica textual entre las dos preguntas de cada par, hay que decidir qué pares están formados por preguntas equivalentes y qué pares no. Por lo tanto la fase de decisión consiste en clasificar los valores finales de similitud semántica textual de cada par de preguntas, entre equivalentes “1” y no equivalentes “0”.

Para llevar a cabo esta decisión es necesario contar con un umbral de decisión. Un umbral de decisión es un valor numérico en el que nos basaremos para decidir qué pares de preguntas albergan preguntas equivalentes semánticamente y que pares no. Es decir, si el valor final de un par de preguntas es superior al umbral de decisión, este par de preguntas será clasificado como equivalente “1”. Por el contrario, si el valor final de un par de preguntas es inferior al umbral de decisión, este par de preguntas será clasificado como no equivalente “0”.

Para llevar a cabo de forma óptima esta decisión es necesario hacerlo con el umbral óptimo. El umbral óptimo es el valor numérico en base al cual decidiremos de forma óptima qué pares de preguntas pertenecen al grupo de los equivalentes “1” y qué pares de preguntas pertenecen al grupo de los no equivalentes “0”, en base a los valores finales obtenidos en la

fase de evaluación. Es decir, el umbral óptimo será el valor numérico que nos permita clasificar los pares de preguntas con el mayor porcentaje de acierto teniendo en cuenta los valores finales obtenidos en la fase de evaluación.

Este porcentaje de acierto es posible calcularlo gracias a la naturaleza del conjunto de datos disponible, ya que éste está debidamente etiquetado. Por lo tanto, para calcular el umbral óptimo utilizaremos un proceso iterativo en el que daremos valores al umbral de decisión y calcularemos sus respectivos porcentajes de acierto. De esta manera, el umbral de decisión que nos permita decidir con un porcentaje de acierto mayor, será el umbral óptimo de decisión para los valores finales obtenidos en la fase de evaluación.

Una vez calculado este umbral de decisión se procederá al cálculo de la similitud textual semántica para los pares de preguntas del conjunto de datos de prueba. Posteriormente, a partir de estos valores calculados en la fase de evaluación y del umbral de decisión, calculado sobre el conjunto de datos de entrenamiento, se decidirá qué pares de preguntas pertenecen al grupo de los equivalentes “1” y qué pares de preguntas pertenecen al grupo de los no equivalentes “0”.

La clasificación obtenida a partir de la decisión anterior será la decisión final del problema planteado y el porcentaje de acierto de esta clasificación nos indicará la fiabilidad de nuestra solución propuesta.

3.3. Tecnologías utilizadas

El procesamiento del lenguaje natural es una rama de la Inteligencia Artificial que estudia la comunicación de las máquinas mediante lenguajes naturales, es decir, estudia técnicas y mecanismos con la finalidad de que una máquina pueda entender e interpretar el lenguaje humano.

El lenguaje de programación Python, ofrece una gran cantidad de herramientas y facilidades para el procesamiento del lenguaje natural y por lo tanto, para el cálculo de similitud textual. A consecuencia de estas prestaciones que ofrece Python, la solución propuesta al problema planteado se ha desarrollado en este lenguaje de programación.

Una de estas facilidades que ofrece Python y que ha sido utilizada para la implementación de la solución del problema propuesto es NLTK (Natural Language Toolkit). NLTK consiste en una plataforma para desarrollar en Python programas orientados al procesamiento del lenguaje natural. Esta plataforma ofrece herramientas tales como corpus lingüísticos y bibliotecas con funcionalidades enfocadas a facilitar el procesamiento de textos.

En el desarrollo de la solución propuesta se han utilizado algunas de las herramientas pertenecientes a NLTK.

En primer lugar se ha utilizado WordNet, una base de datos léxica en inglés. En ella podemos encontrar nombres, adjetivos, verbos y adverbios clasificados en grupos de sinónimos llamados “synsets”. Estos grupos están relacionados entre sí mediante sus

significados semánticos y sus características léxicas, formando así una red de palabras y significados de gran utilidad para el procesado del lenguaje natural. Además de agrupar las palabras por su significado, WordNet clasifica las palabras por su categoría gramatical lo que resulta muy útil a la hora de extraer el significado de una palabra en un contexto concreto.

En la solución propuesta, WordNet ha sido utilizada para comparar los significados semánticos de las palabras de las preguntas de un mismo par con la finalidad de calcular un valor que represente la similitud semántica textual de estas preguntas. Para llevar a cabo estas comparaciones semánticas hay varias posibilidades. Básicamente, WordNet ofrece seis funciones de similitud semántica: Resnik (res) [48], Lin (lin) [49], Jiang & Conrath (jcn) [50], Leacock & Chodorow (lch) [46], Wu & Palmer (wup) [47] y Path length (path) [51].

En segundo lugar se ha utilizado el corpus Stopwords, una base de datos de palabras en inglés que no aportan ningún significado semántico como preposiciones, artículos, determinantes o pronombres. Este corpus nos ha servido en la solución al problema planteado para eliminar todas estas palabras que no aportan significado semántico y así facilitar el cálculo de un valor que represente la similitud semántica de las preguntas de un mismo par.

En último lugar, se han utilizado las herramientas que ofrece NLTK para “tokenizar” y “lematizar” (word tokenize y WordNet lemmatizer). Es muy importante a la hora de procesar textos y en concreto calcular similitudes semánticas, preparar y dar el formato adecuado al conjunto de datos disponible con el objetivo de simplificar y optimizar al máximo estas tareas. Estas herramientas de NLTK nos permiten, por una parte, tokenizar nuestro conjunto de datos, es decir, separar cada palabra en unidades distintas para poder tratarlas por separado y facilitar su comparación, su eliminación o su clasificación. Por otra parte, esta herramienta nos permite lematizar estas palabras previamente tokenizadas, es decir, nos permite reducir a su lema (raíz) las palabras que estén en su forma flexionada, facilitando así la detección de similitud semántica y optimizando el cálculo de similitud semántica textual entre las preguntas de un mismo par.

En la implementación de la solución propuesta también ha sido utilizado Gensim, otro conjunto de herramientas implementado en Python que permite el modelado de espacios vectoriales y el modelado de tópicos. Gracias a Gensim hemos podido utilizar en la solución propuesta Word2Vec, un grupo de modelos relacionados que nos permite producir “word embeddings”, es decir, un espacio vectorial en el que cada palabra es un vector diferente. La distancia entre estos vectores indica la equivalencia semántica de las palabras correspondientes a los mismos. Esta equivalencia es calculada a partir del contexto en el que aparece cada palabra en un corpus. Es decir, Word2Vec recibe como input un corpus de texto y, a partir de este corpus, generará un modelo de tópicos con el que las palabras pueden ser comparadas semánticamente de forma que será capaz de calcular un valor numérico entre cada pareja de palabras que representa la similitud semántica entre ambas. Word2Vec también puede recibir como input un modelo generado previamente y a partir de éste calcular similitudes semánticas entre las palabras que se desee.

A consecuencia de su funcionalidad, Word2Vec ha sido utilizado para comparar las palabras de las preguntas de cada par con el objetivo de calcular un valor numérico que represente la similitud semántica de estas preguntas. El modelo utilizado ha sido proporcionado por el tutor del presente Trabajo de Fin de Grado y fue generado a partir de un algoritmo de deep learning entrenado con artículos de Wikipedia.

Adicionalmente a todas estas herramientas utilizadas en la solución del problema planteado, se ha utilizado Pandas, una librería del lenguaje de programación Python que permite y facilita el análisis de datos, las operaciones con estructuras como tablas numéricas y series temporales o el procesamiento de datos. Pandas es una extensión de NumPy que a su vez es una extensión de Python que le añade soporte para operar con vectores y matrices. En la solución del problema, Pandas ha sido utilizada para facilitar el procesado de los conjuntos de datos, tanto de entrenamiento como de prueba así como para llevar a cabo varias operaciones con vectores y matrices en la fase de evaluación.

4. Experimentos y resultados

Con el objetivo de mostrar la efectividad alcanzada en los métodos implementados durante la realización del presente Trabajo de Fin de Grado, a continuación, se expondrán los experimentos realizados, la metodología y los resultados obtenidos en ellos. Además de mostrar y explicar los resultados obtenidos en el método principal que se corresponde con la solución propuesta al problema planteado, durante este capítulo se muestran y se explican los diferentes resultados obtenidos a partir de todos los métodos implementados.

Por lo tanto, se expondrán y se explicarán por separado los resultados de los métodos que conforman la solución del problema así como los resultados de otros métodos que pese a no formar parte de la solución final también ofrecen unos rendimientos similares.

En primer lugar, y con el objetivo de comprender lo mejor posible el problema propuesto, en este capítulo también se expondrá el corpus completo que ha sido utilizado en la elaboración del presente Trabajo de Fin de Grado, así como sus particiones: el corpus de entrenamiento, con el que se han desarrollado los métodos explicados en el presente Trabajo de Fin de Grado y el corpus de prueba con el que se ha testado la efectividad de dichos métodos. Se detallarán las características más importantes para el diseño de la solución en ambos corpus y se expondrán algunos ejemplos de los pares de preguntas que constituyen estos dos corpus.

4.1. Conjunto de datos

El conjunto de datos utilizado durante el desarrollo de la solución propuesta al problema planteado consiste en 404.290 pares de preguntas en inglés. Estos pares han sido obtenidos del conjunto de entrenamiento facilitado en uno de los retos de Kaggle y están debidamente etiquetados con dos posibilidades: paráfrasis (‘1’) y no paráfrasis (‘0’). De los 404.290 pares de preguntas, 149.263 (36,9%) están etiquetados como paráfrasis (‘1’) mientras que 255.027 (63,1%) están etiquetados como no paráfrasis (‘0’).

Como se ha explicado anteriormente, este conjunto de datos ha sido dividido en dos corpus de datos diferentes, uno destinado al entrenamiento del método a desarrollar y otro destinado a probar el método desarrollado. El conjunto de entrenamiento cuenta con 270.000 pares de preguntas de los cuales 100.368 (37,2%) están clasificados como paráfrasis (‘1’) y 169.632 (62,8%) como no paráfrasis (‘0’). El conjunto de prueba cuenta con 134.290 pares de preguntas de los cuales 48.895 (36,4%) están clasificados como paráfrasis (‘1’) y 85.395 (63,6%) como no paráfrasis (‘0’).

Cantidad de palabras

Una característica importante de este corpus de datos es la cantidad de palabras que tienen las preguntas de los pares que lo conforman. En las palabras de las preguntas es donde

reside la información y el sentido de éstas, por lo que la cantidad de palabras por pregunta nos indicará la cantidad de información que tendremos disponible para estimar un valor que represente la similitud semántica de cada par de preguntas.

En el corpus de datos que tenemos a nuestra disposición la cantidad de palabras por preguntas es muy variada por lo que nuestra solución debe ser adecuada para preguntas cortas y para preguntas largas. La cantidad de palabras aporta información y favorece la precisión de la decisión final, sin embargo, ralentiza la velocidad de procesamiento ya que más operaciones deben ser realizadas.

A continuación se muestran, a través de la tabla 4.1., distintos datos estadísticos sobre la distribución de las palabras en las preguntas de los conjuntos de datos sin preprocesar (Corpus completo, corpus de entrenamiento y corpus de prueba): el número de pares de preguntas, las palabras totales, las palabras diferentes, el número máximo de palabras por pregunta, el número mínimo de palabras por pregunta y la media de palabras por pregunta.

Corpus	**Pares de preguntas**	**Palabras totales**	**Palabras diferentes**	**Máximo de palabras por pregunta**	**Mínimo de palabras por pregunta**	**Media de palabras por pregunta**
Corpus completo	404.290	9.028.149	115.208	249	2	11,06
Corpus de entrenamiento	270.000	6.028.987	93.434	249	2	11,16
Corpus de prueba	134.290	2.999.162	65.439	249	2	11,17

Tabla 4.1. Estadísticas sobre la distribución de palabras en el conjunto de datos sin preprocesar.

Repetición de palabras

Además de la cantidad de palabras por pregunta, otra característica importante es el número de palabras que se repiten en ambas preguntas de un mismo par. Este indicador, no es definitivo pero sí aporta información respecto a la similitud semántica textual que presentan las preguntas de un par.

A continuación se muestran, a través de la tabla 4.2., distintos datos estadísticos sobre la repetición de las palabras en los conjunto de datos sin preprocesar (Corpus completo, corpus de entrenamiento y corpus de prueba): el número de pares de preguntas, las palabras que se repiten, el número máximo de palabras repetidas por pregunta, el número mínimo de palabras repetidas por pregunta y la media de palabras repetidas por pregunta.

Corpus	Pares de preguntas	Palabras repetidas	Máximo de palabras repetidas por pregunta	Mínimo de palabras repetidas por pregunta	Media de palabras repetidas por pregunta
Corpus completo	404.290	1.998.352	40	0	4,94
Corpus de entrenamiento	270.000	1.335.425	38	0	4,95
Corpus de prueba	134.290	662.927	40	0	4,94

Tabla 4.2. Estadísticas sobre la repetición de palabras entre los pares del conjunto de datos sin preprocesar.

Repetición de preguntas

Otra característica importante de este corpus de datos que tenemos disponible es la repetición de una gran parte de las preguntas de los pares del propio corpus. Es decir, dentro del propio conjunto de datos, podemos encontrar preguntas que se repiten. A continuación se adjuntan tres gráficas (Figuras 4.1., 4.2. y 4.3.) donde se pueden observar el número de veces que se repiten ciertas preguntas y el número de preguntas por cada cantidad de veces repetidas en el corpus de datos completo, en el corpus de entrenamiento y en el corpus de prueba.

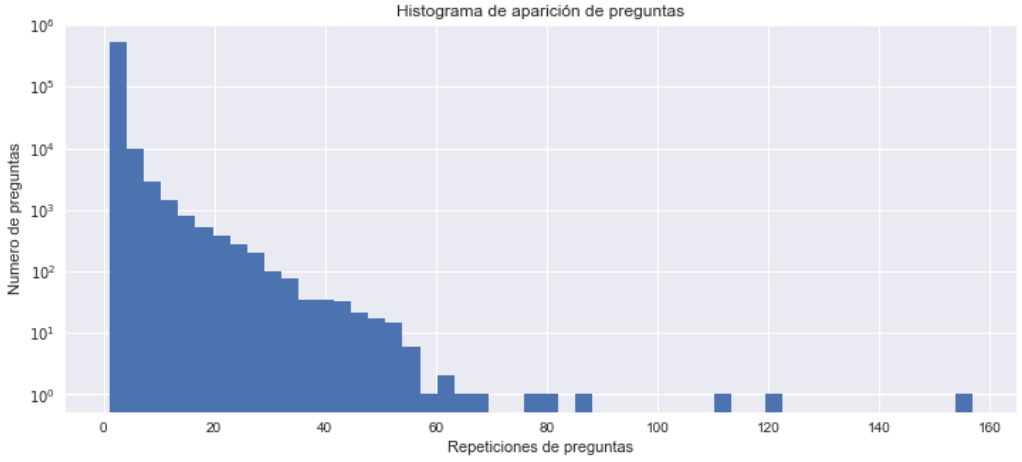

Figura 4.1. Histograma de aparición de las preguntas del corpus de datos completo

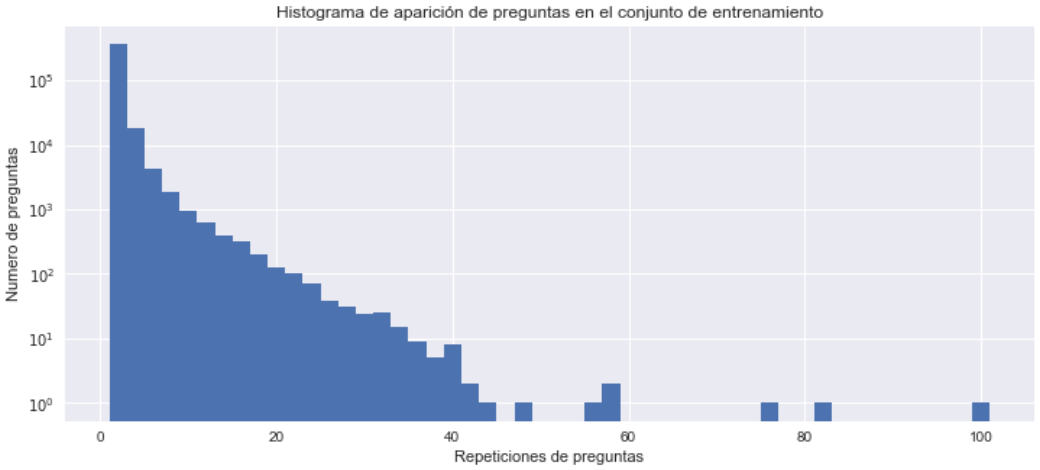

Figura 4.2. Histograma de aparición de las preguntas del corpus de entrenamiento.

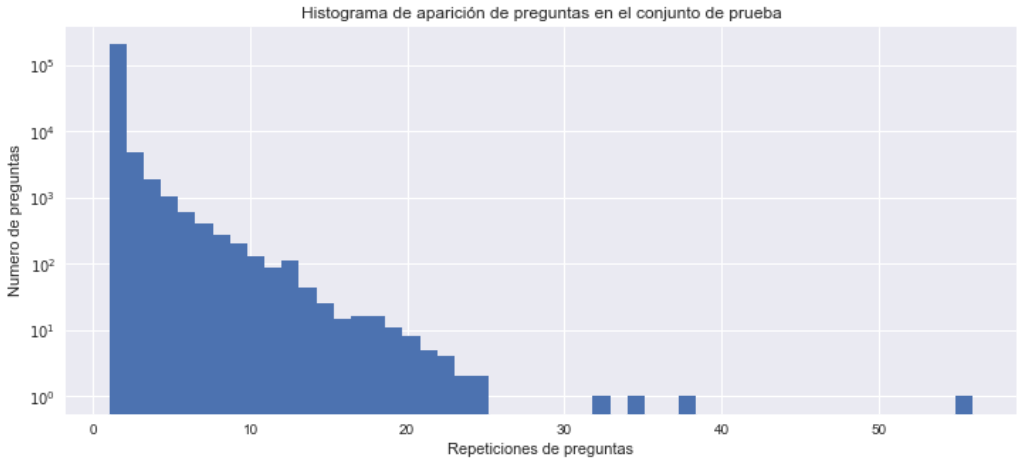

Figura 4.3. Histograma de aparición de las preguntas del corpus de prueba.

En estas gráficas podemos observar que la mayoría de preguntas del corpus de datos no aparecen repetidas o aparecen repetidas pocas veces, sin embargo existe una cantidad notable de preguntas repetidas en el corpus de datos. En las gráficas se observa que el número de preguntas disminuye conforme aumenta el número de veces que se repiten las preguntas, es decir, en el corpus de datos hay pocas preguntas que se repiten muchas veces y hay muchas preguntas que aparecen una o pocas veces. Las tres gráficas tienen una distribución similar por lo que podemos deducir que las preguntas del corpus están uniformemente repartidas en el corpus de entrenamiento y en el corpus de prueba.

Ejemplos

Con el objetivo de conocer mejor el corpus de datos y tener una mejor visión general del problema, a continuación se expondrán doce ejemplos de pares de preguntas que presentan características diferentes. Además de los ejemplos de los pares de preguntas, se ha añadido el etiquetado original, es decir, la solución a la pregunta de la existencia de

paráfrasis. Los pares evaluados como “1” presentan paráfrasis y los evaluados como “0” no presentan paráfrasis.

- *Ejemplo 1. Par 49º del corpus de datos:*
    - *"How do I make friends."*
    - *"How to make friends ?"*

Evaluado como "1".

- *Ejemplo 2. Par 52º del corpus de datos:*
    - *"Nd she is always sad?"*
    - *"Aerodynamically what happens when propellor rotates?"*

Evaluado como "0".

- *Ejemplo 3. Par 53º del corpus de datos:*
    - *"What is the best/most memorable thing you've ever eaten and why?"*
    - *"What is the most delicious dish you've ever eaten and why?"*

Evaluado como "1".

- *Ejemplo 4. Par 168630º del corpus de datos:*
    - *"Why is change management important in healthcare?"*
    - *"Why is change management important?"*

Evaluado como "0".

- *Ejemplo 5. Par 168824º del corpus de datos:*
    - *"What is the possibility of war between India and Pakistan after surgical operation?"*
    - *"How many people think that India is heading toward war with pak after surgical strike?"*

Evaluado como "1".

- *Ejemplo 6. Par 256617º del corpus de datos:*
    - *"What are some ways to describe landscapes?"*
    - *"How do I describe a good landscape?"*

Evaluado como "1".

- *Ejemplo 7. Par 282789º del corpus de datos:*
    - *"What is the pen?"*
    - *"What is a G-pen?"*

Evaluado como "0".

- *Ejemplo 8. Par 347726º del corpus de datos:*
    - *"If J can paint a room in 6 hours, and T can paint the same room in 12 hours, how long would it take J & T to paint the room together?"*
    - *"If you are going 100 miles per hour, how long would it take you to travel 1 mile?"*

Evaluado como "0".

- *Ejemplo 9. Par 391717º del corpus de datos:*
    - *"Will image quality be worse if my lense has more glass?"*
    - *"Are CCD/CMOS and image processing technology more important than the quality of the lens to produce a good image or video?"*

Evaluado como "0".

- *Ejemplo 10. Par 343548º del corpus de datos:*
    - *"How do I reduce my tummy without doing any exercise?"*
    - *"How can I lose belly fat without any exercise?"*

Evaluado como "1".

En estos diez primeros ejemplos de pares de preguntas a evaluar, podemos observar características diferentes entre ellos. Básicamente se pueden diferenciar cuatro clases de pares de preguntas en cuanto a la repetición literal de palabras: los que contienen preguntas que comparten muchas palabras y están etiquetados como “1”, los que contienen preguntas con pocas palabras en común y también están etiquetados como “1” y las dos clases restantes estarían formadas por las de las mismas características que las anteriores pero etiquetadas como “0”.

De esta manera podemos observar que la simple comparación literal de palabras pese a aportar cierta información de similitud semántica textual, no es definitiva y que es necesario extraer información semántica de las preguntas para alcanzar una decisión, en cuanto a la paráfrasis, más precisa.

- *Ejemplo 11. Par 18055º del corpus de datos:*

    - *"Im moving to NY. My Dr gave me 2 refills of Xanax, but pharmacy said by law, they couldn't give me more than 1 refill per month. Is it true?"*
    - *"Heartbreak? Heartbreak? She's my girlfriend for two months, I chose her over my girlfriend for 2 years. I like her so much to the point that I can't let her go even if she wants to end our relationship because of the other people around us most especially her family. I do the things for her that I'm not used to for a girl and I am willing to sacrifice everything just to have a little time with her. A little and limited time that I'm asking from her but she don't wanna give it to me. She's scared that someone might see us, that she's still having an affair with me. I love her and I want to be with her at least once a week even if just for a limited time. I'm not sure if I'm doing the right thing, all of my friends told me to stop it and just let it go 2 months is just 2 months not a deep relationship. But they don't feel what I feel, in this span of time I learned a lot, I learned how to love, to be loved, to sacrifice a good life, and to sacrifice a better clear future. My mind tells me to stop, but my heart tells me to hold, don't give up, stay with her and give her the unconditional love. Should I follow my mind or follow my heart?"*

Evaluado como "0".

- *Ejemplo 12. Par 8361º del corpus de datos:*
    - *"Cloud certification?"*
    - *"How do you show whether 2^n+5^n+2^(n+1) +5^(n+1) is divisible by 3?"*

Evaluado como "0".

En estos dos últimos ejemplos de pares de preguntas podemos observar principalmente la diferencia de longitud de palabras que existe entre las preguntas de los distintos pares del corpus de datos. Como se ha explicado anteriormente las palabras aportan información semántica muy útil a la hora de decidir la existencia de paráfrasis, pero cuando las dos preguntas a comparar presentan una cantidad de palabras muy distinta, la comparación literal de palabras o la comparación semántica de éstas puede reportarnos valores poco fiables. Esto ocurre porque pese a que ambas preguntas presentan palabras equivalentes o similares, presentan una cantidad mayor de palabras diferentes que es conveniente tener en cuenta a la hora de calcular un valor que represente la similitud semántica entre ambas preguntas de cada par.

Preprocesado

Antes de comenzar a realizar operaciones y calcular valores representativos de similitud semántica textual con los pares de preguntas del conjunto de datos, estos deben ser preprocesados. Es decir, los diferentes pares del conjunto de datos deben ser sometidos a varios procesos que los adecuarán para facilitar las operaciones y los cálculos que se efectuarán posteriormente. Este preprocesado ha sido explicado anteriormente en el

capítulo 3, pero a continuación se mostrará el cambio que presentan los ejemplos mostrados anteriormente así como la variación en las estadísticas presentadas al principio de esta sección.

Cantidad de palabras

Una de las características que varía en las preguntas del corpus de datos, es la cantidad de palabras ya que muchas de éstas son eliminadas por no aportar información semántica o son transformadas para facilitar su comparación.

A continuación se muestran, a través de la tabla 4.3., distintos datos estadísticos sobre la distribución de las palabras en las preguntas de los conjuntos de datos una vez han sido preprocesados (Corpus completo, corpus de entrenamiento y corpus de prueba): el número de pares de preguntas, las palabras totales, las palabras diferentes, el número máximo de palabras por pregunta, el número mínimo de palabras por pregunta y la media de palabras por pregunta.

Corpus	**Pares de preguntas**	**Palabras totales**	**Palabras diferentes**	**Máximo de palabras por pregunta**	**Mínimo de palabras por pregunta**	**Media de palabras por pregunta**
Corpus completo	404.290	4.542.054	91.979	109	1	5,62
Corpus de entrenamiento	270.000	3.032.547	73.679	109	1	5,62
Corpus de prueba	134.290	1.509.507	50.671	109	1	5,62

Tabla 4.3. Estadísticas sobre la distribución de palabras en el conjunto de datos preprocesado.

Repetición de palabras

Otra de las características que varía tras el preprocesado de los datos es el número de palabras que se repiten entre las preguntas de los pares.

A continuación se muestran, a través de la tabla 4.4., distintos datos estadísticos sobre la repetición de las palabras en los conjuntos de datos una vez han sido preprocesados (Corpus completo, corpus de entrenamiento y corpus de prueba): el número de pares de preguntas, las palabras que se repiten, el número máximo de palabras repetidas por pregunta, el número mínimo de palabras repetidas por pregunta y la media de palabras repetidas por pregunta.

Corpus	Pares de preguntas	Palabras repetidas	Máximo de palabras repetidas por pregunta	Mínimo de palabras repetidas por pregunta	Media de palabras repetidas por pregunta
Corpus completo	404.290	1.104.830	24	0	2,73
Corpus de entrenamiento	270.000	738.434	23	0	2,73
Corpus de prueba	134.290	366.396	24	0	2,73

Tabla 4.4. Estadísticas sobre la repetición de palabras entre los pares del conjunto de datos preprocesado.

Palabras más frecuentes

Después de haber preprocesado los datos del corpus se ha realizado un análisis sobre la aparición de las diferentes palabras en el corpus de datos. Este análisis adquiere sentido una vez los datos han sido preprocesados ya que la frecuencia de aparición de las palabras en el corpus sin preprocesar no nos aporta información a la hora de tener una visión más completa de los datos con los que se va a contar para decidir la existencia de paráfrasis en los pares de preguntas.

Este análisis sobre la aparición de las palabras más frecuentes del corpus de datos aporta información sobre la distribución de las palabras dentro del corpus y nos da una visión más amplia del corpus que tenemos a nuestra disposición. A continuación se muestran tres histogramas (Figuras 4.4., 4.5. y 4.6.) donde se pueden apreciar las palabras más frecuentes y las veces que éstas se repiten a lo largo del corpus completo de datos, del corpus de entrenamiento y del corpus de prueba:

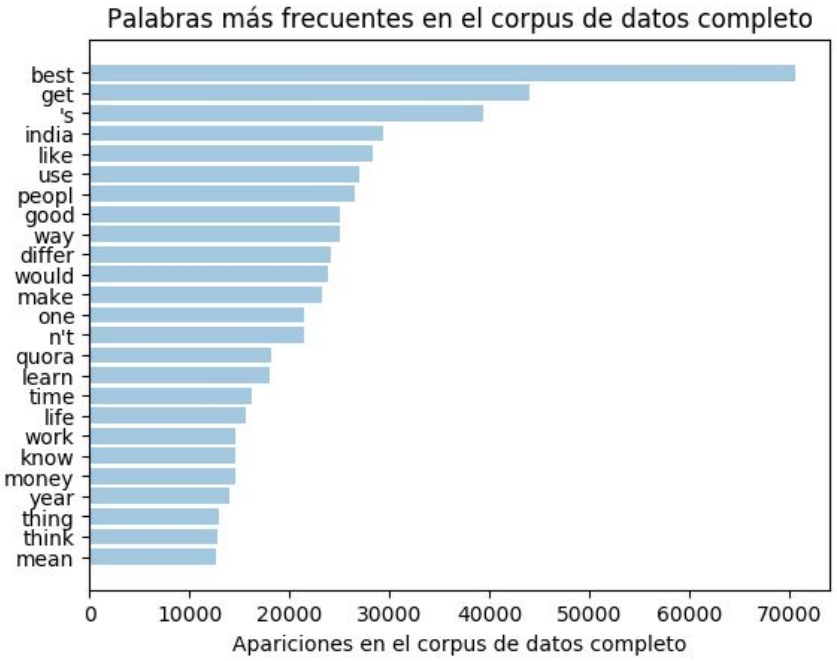

Figura 4.4. Histograma de las palabras más frecuentes del corpus de datos completo

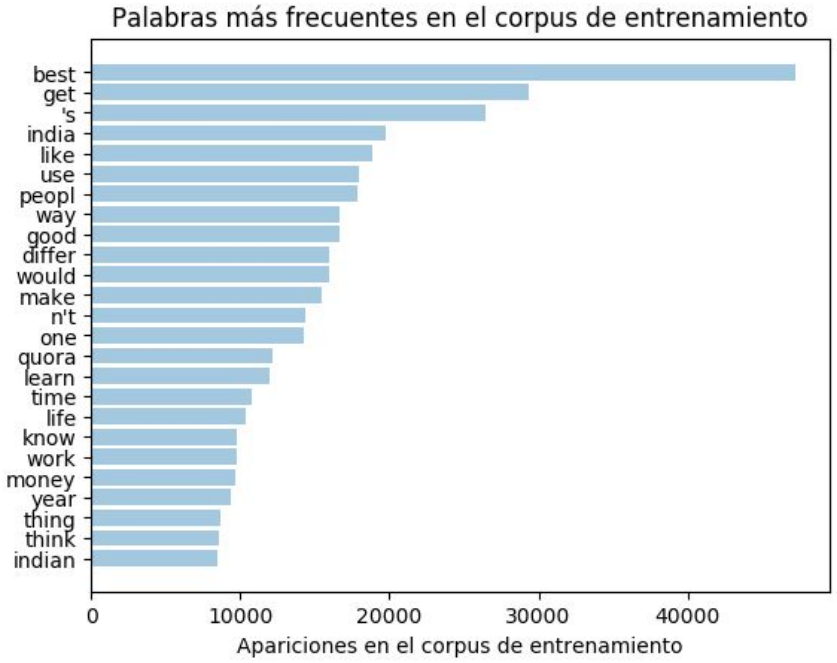

Figura 4.5. Histograma de las palabras más frecuentes del corpus de entrenamiento

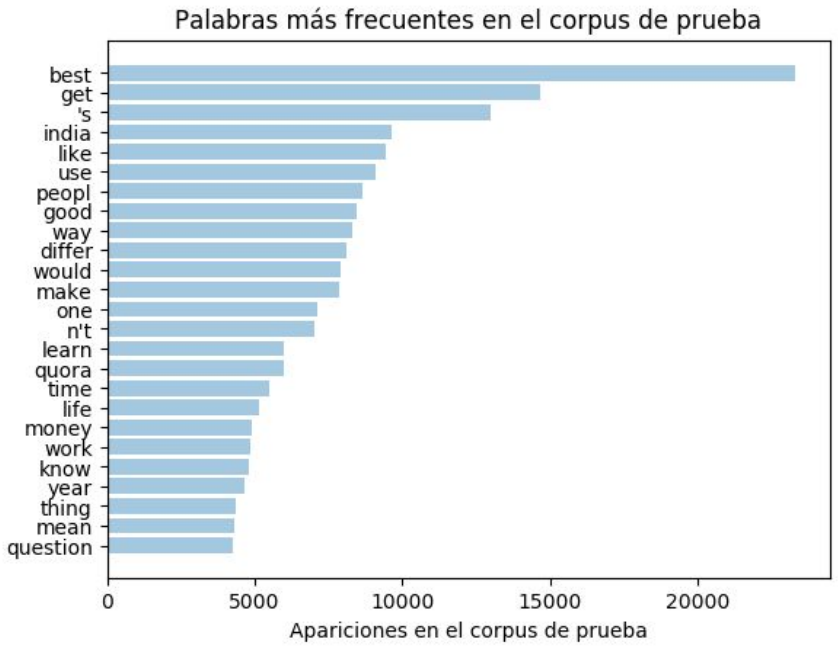

Figura 4.6. Histograma de las palabras más frecuentes del corpus de prueba

En estos histogramas podemos observar que la distribución de las palabras en los corpus de entrenamiento y de prueba es similar a la del corpus completo de datos por lo que ambos subcorpus son conjuntos representativos del total y similares entre sí. Esta característica es favorable a la hora de entrenar y de evaluar la solución al problema ya que nos ayudará a obtener un resultado final más preciso y fiable.

Ejemplos

A continuación se van a presentar los mismos ejemplos de pares de frases que han sido presentados anteriormente, pero tras el preprocesado de datos. Es decir, se presentarán los pares de preguntas una vez han sido preprocesados con el método que ha sido implementado en la solución del presente Trabajo de Fin de Grado.

- *Ejemplo 1. Par 49º del corpus de datos:*

    - *“make”, “friend”*
    - *“make”, “friend”*

    Evaluado como “1”

- *Ejemplo 2. Par 52º del corpus de datos:*

    - *“nd”, “alway”, “sad”*
    - *“aerodynam”, “happen”, “propellor”, “rotat”*

Evaluado como “0”

- *Ejemplo 3. Par 53º del corpus de datos:*

    - *"best/most”, “memor”, “thing”, “ve”, “ever”, “eaten”*
    - *"delici”, “dish”, “ve”, “ever”, “eaten”*

Evaluado como "1".

- *Ejemplo 4. Par 168630º del corpus de datos:*

    - *“chang”, “manag”, “import”, “healthcar”*
    - *“chang”, “manag”, “import”*

Evaluado como "0".

- *Ejemplo 5. Par 168824º del corpus de datos:*

    - *“possibl”, “war”, “India”, “pakistan”, “surgic”, “oper”*
    - *“mani”, “peopl”, “think”, “india”, “head”, “toward”, “war”, “pak”, “surgic”, “strike”*

Evaluado como "1".

- *Ejemplo 6. Par 256617º del corpus de datos:*

    - *“way”, “describ”, “landscap”*
    - *“describ”, “good”, “landscap”*

Evaluado como "1".

- *Ejemplo 7. Par 282789º del corpus de datos:*

    - *“pen”*
    - *“g-pen”*

Evaluado como "0".

- *Ejemplo 8. Par 347726º del corpus de datos:*

    - *“j”, “paint”, “room”, “6”, “hour”, “paint”, “room”, “12”, “hour”, “long”, “would”, “take”, “j”, “paint”, “room”, “togeth”*
    - *“go”, “100”, “mile”, “per”, “hour”, “long”, “would”, “take”, “travel”, “1”, “mile”*

Evaluado como "0".

- *Ejemplo 9. Par 391717º del corpus de datos:*

- *“imag”, “qualiti”, “wors”, “lens”, “glass”*
- *“ccd/cmos”, “imag”, “process”, “technolog”, “import”, “qualiti”, “len”, “produc”, “good”, “imag”, “video”*

Evaluado como "0".

- *Ejemplo 10. Par 343548º del corpus de datos:*

  - *“reduc”, “tummi”, “without”, “exercis”*
  - *“lose”, “belli”, “fat”, “without”, “exercis”*

Evaluado como "1".

- *Ejemplo 11. Par 18055º del corpus de datos:*

  - *“Im”, “move”, “ny”, “dr”, “gave”, “2”, “refil”, “xanax”, “pharmaci”, “said”, “law”, “could”, “n't”, “give”, “1”, “refil”, “per”, “month”, “true”*
  - *“heartbreak”, “heartbreak”, “'s”, “girlfriend”, “two”, “month”, “chose”, “girlfriend”, “2”, “year”, “like”, “much”, “point”, “ca”, “n't”, “let”, “go”, “even”, “want”, “end”, “relationship”, “peopl”, “around”, “us”, “especi”, “famili”, “thing”, “'m”, “use”, “girl”, “will”, “sacrific”, “everyth”, “littl”, “time”, “littl”, “limit”, “time”, “'m”, “ask”, “n't”, “wan”, “na”, “give”, “'s”, “scare”, “someon”, “might”, “see”, “us”, “'s”, “still”, “affair”, “love”, “want”, “least”, “week”, “even”, “limit”, “time”, “'m”, “sure”, “'m”, “right”, “thing”, “friend”, “told”, “stop”, “let”, “go”, “2”, “month”, “2”, “month”, “deep”, “relationship”, “n't”, “feel”, “feel”, “span”, “time”, “learn”, “lot”, “learn”, “love”, “love”, “sacrific”, “good”, “life”, “sacrific”, “better”, “clear”, “futur”, “mind”, “tell”, “stop”, “heart”, “tell”, “hold”, “n't”, “give”, “stay”, “give”, “uncondit”, “love”, “follow”, “mind”, “follow”, “heart”*

Evaluado como "0".

- *Ejemplo 12. Par 8361º del corpus de datos:*
  - *“cloud”, “certif”*
  - *“show”, “whether”, “2^n+5^n+2^”, “(n+1)”, “+5^”, “(n+1)”, “divisiblev”, “3”*

Evaluado como "0".

Observando estos ejemplos y su evolución respecto a la primera muestra de ellos sin preprocesar se pueden apreciar los cambios en la cantidad de palabras y en ciertas terminaciones de palabras que favorecen la comparación de las preguntas y la extracción de la información semántica de éstas.

4.2. Experimentos y Resultados

Con el objetivo de evaluar los métodos implementados durante la elaboración del presente Trabajo de Fin de Grado, a continuación se presentan los resultados de los experimentos realizados con los métodos mencionados.

Para obtener los resultados de los experimentos realizados con los distintos métodos implementados en el presente Trabajo de Fin de Grado se ha aplicado la misma metodología. Esta metodología se ha descrito en el *Capítulo 3. Problema planteado y solución propuesta,* mediante el diagrama de bloques representado en la *Figura 3.1*. En este diagrama de bloques se muestran las diferentes fases en las que está dividida la solución al problema planteado. Este diagrama de bloques es aplicable a cualquiera de los experimentos que se presentan a continuación.

Para comparar la efectividad de los diferentes métodos, se expondrán las métricas estándar derivadas de la matriz de confusión: Exactitud y Medida F. Además se explicará la intención de cada experimento realizado, los resultados obtenidos y se expondrán las conclusiones correspondientes.

Experimento 1. Métodos basados en la repetición de palabras: Comparador literal de palabras y Método de conteo.

En este experimento se probarán el Método 1. Comparador literal de palabras y el Método 2. Método de conteo. En estos métodos, como se ha explicado durante el Capítulo 3, se utilizan métricas basadas en la repetición de palabras con el objetivo de detectar la similitud semántica entre las preguntas de cada par. En el Método 1. Comparador literal de palabras, para obtener el valor representativo de la similitud semántica se han combinado mediante una media aritmética las siguientes métricas: coeficiente de Dice [37], Coeficiente de Jaccard [38], Coeficiente de traslape [39] y Coeficiente del Coseno [40]. En el Método 2. Método de conteo, se ha utilizado una metodología particular que ha sido explicada durante el Capítulo 3, 3.2.2. Evaluación.

Los objetivos de este experimento son evidenciar la alta relación de efectividad y tiempo de procesamiento de los métodos que implementan técnicas basadas en el número de palabras repetidas así como comparar el rendimiento que presentan estos métodos basados en la repetición de las palabras.

Mientras que los métodos 3, 4 y 5 (Comparador semántico de palabras, Máxima similitud y Método híbrido) presentan una velocidad de procesamiento similar, los métodos 1 y 2 (Comparador literal de palabras y Método de conteo) presentan unas medidas de efectividad y medida F inferiores. No obstante, presentan una velocidad de procesamiento cuatro veces mayor a la de los métodos anteriores, por lo que se puede afirmar que la relación entre la efectividad y el tiempo de procesamiento es alta.

Método	Exactitud	Medida F
Comparador literal de palabras (media aritmética)	**0.676**	**0.265**
Coeficiente de Dice	0.659	0.247
Coeficiente de Jaccard	0.661	0.25
Coeficiente de Traslape	0.663	0.252
Coeficiente de Coseno	0.657	0.244
Método de conteo	0.659	0.246

Tabla 4.5.. Exactitud y Medida F para el Método 1. Comparador literal de palabras y Método 2. Método de Conteo

Los resultados obtenidos en este experimento ponen de manifiesto la efectividad de los métodos utilizados teniendo en cuenta su alta velocidad de procesamiento. Se puede observar que el Método 1. Comparador literal de palabras presenta una eficiencia y una medida F mayores que el Método 2. Método de conteo, esto se debe a que la combinación de varias métricas aumenta la precisión del método que las implemente frente a los métodos que se basan en una sola métrica con el objetivo de calcular un valor que represente la similitud semántica textual.

Experimento 2. Métodos basados en la similitud semántica de palabras: Comparador semántico de palabras y Máxima similitud.

En este experimento se probarán el Método 3. Comparador semántico de palabras y el Método 4. Máxima similitud. En estos métodos, como se ha explicado durante el Capítulo 3 se utilizan métricas que calculan la similitud semántica entre las palabras de las preguntas de cada par con el objetivo de detectar la similitud semántica entre las preguntas de los pares del conjunto de datos. En el Método 3. Comparador semántico de palabras, con el fin de obtener un valor que represente la similitud semántica entre las preguntas de cada par, se utilizan todas las medidas de similitud semántica entre las palabras de la pregunta 1 y la pregunta 2. Mientras que en el Método 4. Máxima similitud, se utiliza solamente la máxima medida de similitud que presenta cada palabra con las palabras de la otra pregunta.

Los objetivos de este experimento son, en primer lugar, evidenciar la mejora que suponen las técnicas basadas en la similitud semántica de las palabras frente a las técnicas basadas en la repetición de palabras y en segundo lugar, se comparará la efectividad de estos dos métodos basados en la similitud semántica de las palabras.

Método	Exactitud	Medida F
Comparador semántico de palabras	0.689	0.277
Medida Wu & Palmer	0.683	0.269
Medida Path lenght	0.685	0.271
Medida Jiang & Conrath	0.687	0.276
Medida Word2Vec	0.687	0.275
Máxima similitud	**0.691**	**0.279**
Medida Wu & Palmer	0.685	0.27
Medida Path lenght	0.686	0.275
Medida Jiang & Conrath	0.688	0.276
Medida Word2Vec	0.69	0.279

Tabla 4.6.. Exactitud y Medida F para el Método 3. Comparador semántico de palabras y Método 4. Máxima similitud.

Los resultados obtenidos en este experimento demuestran que el uso de técnicas basadas en la similitud semántica de palabras presenta mejores prestaciones que el uso de las técnicas basadas en la repetición de palabras. Estos resultados se deben a la capacidad de las técnicas basadas en la similitud semántica de palabras que, además de tener en cuenta la información referente a las palabras repetidas entre ambas preguntas, son capaces de identificar y extraer información sobre las relaciones semánticas de las palabras de ambas preguntas. La identificación y extracción de esta información es de suma importancia en la detección de paráfrasis ya que esta consiste en la equivalencia semántica de los textos. No obstante, los resultados ofrecidos por estos dos métodos que utilizan técnicas basadas en la similitud semántica de las palabras no suponen una mejora notable respecto al rendimiento ofrecido por los métodos anteriores, basados en la repetición de palabras.

Además, estos resultados ponen de manifiesto que ambos métodos presentan un rendimiento similar. Aunque el Método 4. Máxima similitud presenta unos resultados ligeramente más precisos, ambos métodos pueden ser considerados similares en cuanto a precisión. Este hecho se debe a que tras ser aplicado el preprocesado del texto, se reduce notablemente el número de palabras por pregunta, por lo que las diferencias de las características de estos dos métodos no tienen tanta repercusión en el resultado final. Es decir, dado que hay menos palabras en las preguntas, la diferencia en la decisión final en el caso de tener en cuenta todas las medidas de similitud entre las palabras (Método 3. Comparador semántico de palabras) y en el caso de tener en cuenta solamente las

máximas similitudes entre las palabras de las preguntas (Método 4. Máxima similitud) disminuye notablemente.

Experimento 3. Método que combina medidas de similitud léxica y medidas de similitud semántica: Método híbrido.

En este experimento se probará el Método 5. Método híbrido. Este método combina técnicas basadas en la similitud semántica de las palabras con técnicas basadas en la repetición de las palabras entre las preguntas de cada par. Además introduce una técnica basada en la disimilitud semántica con el fin de aumentar la diferencia entre los valores representativos de similitud semántica correspondientes a los pares en los que hay paráfrasis y los valores representativos de similitud semántica correspondientes a los pares en los que no.

El objetivo de este experimento es comparar el rendimiento de este método, que combina métricas de distintas naturalezas (similitud semántica y repetición de palabras) con el rendimiento que ofrece la combinación de los cuatro métodos anteriores mediante la media aritmética de los valores representativos de la similitud semántica que devuelven. Con esta comparación se decidirá cual de estos dos métodos es el propuesto para la solución del presente Trabajo de Fin de Grado.

Método	**Exactitud**	**Medida F**
Método híbrido	**0.706**	**0.286**
Media aritmética (Métodos 1-4)	0.694	0.282

Tabla 4.7. Exactitud y Medida F para el Método 5. Método híbrido y para la media aritmética de los cuatro primeros métodos.

Los resultados obtenidos en este experimento ponen de manifiesto que el Método híbrido es más efectivo que la media aritmética de los resultados obtenidos entre los otros cuatro métodos. De este resultado se puede concluir que la técnica basada en la disimilitud de las palabras de ambas preguntas aumenta la diferencia entre los valores calculados correspondientes a los pares cuyas preguntas son equivalentes semánticamente y los valores calculados correspondientes a los pares cuyas preguntas no son equivalentes semánticamente.

En base a estos resultados se ha decidido que el Método 5. Método híbrido, es el método propuesto como solución para el problema planteado en el presente Trabajo de Fin de Grado.

A causa de que el problema planteado en el presente Trabajo de Fin de Grado está inspirado en una competición de la plataforma Kaggle, se valoró incluir una comparativa entre los resultados obtenidos con los métodos propuestos en el presente Trabajo de Fin de Grado con los principales resultados de la competición mencionada. Sin embargo, la clasificación de los resultados de la competición de Kaggle está hecha en base a un 'score' del que no se dispone el detalle de su cálculo ni el valor que representa. Por lo tanto, no ha sido posible llevar a cabo dicha comparativa.

5. Marco regulador

5.1. Legislación aplicable

El desarrollo del presente Trabajo de Fin de Grado ha sido llevado a cabo en el lenguaje de programación Python 3. La licencia vigente que regula Python 3 es Python Software Foundation License (PSFL) [69]. PSFL es considerada una licencia de software libre ya que cumple con los requisitos establecidos por Open Source Initiative (OSI) [70]. PSFL al igual que la mayoría de licencias de Python, es GPL-compatible. Sin embargo, PSFL no se trata de una licencia *copyleft* [71] y permite modificar el código fuente, reproducir, analizar, probar, crear, mostrar públicamente, desarrollar trabajos y distribuirlos sin que estos tengan que estar clasificados como trabajos de código abierto.

5.2. Estándares técnicos

La solución expuesta en el presente Trabajo de Fin de Grado está desarrollada en su totalidad con el lenguaje de programación de Python. El estándar técnico que propone Python para escribir código es PEP 8 [72] y en él se describen las pautas recomendadas para generar un código legible y fácil de comprender.

No obstante, la solución del problema se ha enfocado desde un punto de vista más analítico y de investigación, por lo que se ha restado importancia al aspecto formal del código y se le ha dado más peso a estos aspectos de investigación y de análisis de los datos facilitados.

Por lo tanto, en el código que constituye la solución del presente Trabajo de Fin de Grado no se ha aplicado completamente el estándar PEP 8. Sin embargo, el código presenta un estilo legible y ordenado en el que se pueden leer comentarios explicativos de las funcionalidades de los distintos bloques de código.

5.3. Propiedad intelectual

En el presente Trabajo de Fin de Grado, no va a ser protegida la propiedad intelectual ya que como las técnicas implementadas en la solución del problema propuesto no han sido ideadas en el presente Trabajo de Fin de Grado, la solución no constituye un producto con posibilidad de ser patentado. Además, el objetivo del trabajo no es el de crear un producto comercial, sino el de investigar y analizar un problema de gran importancia en la actualidad con el fin último de aportar ideas que puedan mejorar las soluciones implementadas o las soluciones que se implementen en trabajos futuros.

6. Entorno socio-económico

6.1. Presupuesto

A continuación, se detalla el presupuesto del presente Trabajo de Fin de Grado. En él se especifican los distintos recursos que han sido utilizados durante su realización.

El precio de las horas del alumno se ha obtenido del sitio web, TuSalario.es [73] donde es posible calcular el salario estimado en función de la formación y la experiencia del trabajador. En este caso, para calcular el precio de la hora trabajada en este Trabajo de Fin de Grado, se ha seleccionado el salario de un Graduado en Ingeniería de sistemas audiovisuales sin experiencia previa. Con estas características de formación y experiencia el precio de la hora trabajada es de 10,5 €.

El presente Trabajo de Fin de Grado se inició el día 1 de febrero de 2017 y se finalizó el día 19 de junio de 2018, un total de 17 meses. A partir de septiembre de 2017, la elaboración del presente Trabajo de Fin de Grado tuvo que ser compaginada con un trabajo a jornada completa, por lo que el tiempo de dedicación se vio reducido y el tiempo de finalización se extendió hasta la fecha especificada anteriormente. El tiempo medio dedicado al mes por el alumno, desde el comienzo del presente Trabajo de Fin de Grado hasta el último mes, fue de 25 horas. En total el tiempo dedicado al desarrollo del presente Trabajo de Fin de Grado es de: 425 horas.

Para implementar la solución del presente Trabajo de Fin de Grado se ha utilizado un ordenador portátil Toshiba.

A continuación, en la Tabla 6.1. se detallan los recursos utilizados durante la ejecución del presente Trabajo de Fin de Grado y sus costes correspondientes:

Recurso utilizado	Coste	Vida útil	Tiempo de uso	Coste estimado
Graduado en Ingeniería de sistemas audiovisuales	10,5 €/hora	NA	425 horas	4.462,5 €
Ordenador portátil Toshiba	700 €	60 meses	17 meses	198,3 €
TOTAL				**4.660,8 €**

Tabla 6.1. Presupuesto de los recursos utilizados para la ejecución del presente Trabajo de Fin de Grado.

6.2. Impacto socio-económico

En el presente Trabajo de Fin de Grado se propone la solución para un problema de detección de preguntas repetidas. El problema planteado está inspirado en un reto de Kaggle (www.kaggle.com), una web de competiciones relacionadas con las ciencias de la computación, en la que se facilitan recursos muy útiles para la implementación de la solución al problema, como un conjunto de datos sobre el que trabajar o un entorno online para desarrollar el código.

A causa de estas características del problema, la solución propuesta presenta un carácter educativo y no comercial. Por lo que esta no tendría ningún impacto socio-económico en la sociedad.

No obstante, la solución propuesta al problema planteado en el presente Trabajo de Fin de Grado puede ser utilizada en numerosas aplicaciones:

- Evaluación de traducción automática [16].
- Clasificación de documentos [4].
- Evaluación de resúmenes redactados automáticamente [18].
- Detección de paráfrasis [19].
- Sugerencia de respuestas automáticas [20].

Otras aplicaciones más concretas cuyo impacto socio-económico ha sido analizado porque fueron consideradas para el enfoque principal del presente Trabajo de Fin de Grado son: la detección de preguntas repetidas en el Congreso de los Diputados y la detección de tuits repetidos con el objetivo de encontrar plagiadores en Twitter.

Todas estas aplicaciones ya han sido implementadas y presentan un rendimiento efectivo por lo que el impacto socio-económico de la solución propuesta en este Trabajo de Fin de Grado en alguna de estas aplicaciones no es considerable. Adicionalmente, las aplicaciones más modernas que están siendo desarrolladas en la actualidad, implementan ideas basadas en aprendizaje automático supervisado cuyo rendimiento es superior a las técnicas utilizadas en el presente Trabajo de Fin de Grado. Por esta razón el impacto socio-económico que pudiera tener una aplicación que utilizase el método desarrollado en el presente Trabajo de Fin de Grado no será considerable.

Sin embargo, la solución implementada en el presente Trabajo de Fin de Grado puede ser un punto de partida interesante para implementar alguna de las aplicaciones citadas anteriormente. Teniendo en cuenta las técnicas implementadas en el presente Trabajo de Fin de Grado y combinándolas con técnicas más modernas como las basadas en aprendizaje automático supervisado que ya presentan un rendimiento considerable [13], sería posible alcanzar un impacto socio-económico reseñable.

7. Conclusiones y trabajo a futuro

A continuación se expondrán las conclusiones obtenidas durante la realización del presente Trabajo de Fin de Grado y se explicarán las líneas de trabajo a futuro identificadas para enriquecer la solución propuesta.

7.1. Conclusiones

Como se ha podido observar en los resultados obtenidos, la efectividad de los métodos propuestos aumenta conforme éstos combinan un mayor número de técnicas de diferentes naturalezas. En primer lugar, se desarrollaron métodos que solo tenían en cuenta la repetición de palabras para calcular similitud semántica textual. Después, se desarrollaron métodos que añadían técnicas basadas en la similitud semántica de las palabras para capturar una mayor cantidad de información y el rendimiento de estos aumentó. Finalmente, se diseñó un método que combinaba estas técnicas e incluía otra más basada en la disimilitud y su efectividad superó a la de los métodos anteriores.

En base a la efectividad que presenta este último método (Método 5. Método híbrido) que incluso supera la efectividad presentada por el método que combina los valores calculados por los cuatro primeros métodos, se puede concluir que la medida de disimilitud que implementa es efectiva y que la inclusión de técnicas diferentes que sean capaces de capturar distintos tipos de información referente a la paráfrasis enriquece y aumenta la precisión de la solución.

A pesar de esta mejora progresiva relacionada con el incremento de técnicas de distintas naturalezas, la diferencia de efectividad alcanzada entre los distintos métodos no es notable. Por esto, se puede afirmar que los métodos que se basan en la repetición de preguntas presentan un buen rendimiento teniendo en cuenta que el tiempo de procesamiento que necesitan es cuatro veces menor al que necesitan el resto de métodos propuestos.

Al observar los resultados obtenidos se puede concluir que estos no suponen una solución definitiva al problema propuesto. Si bien es cierto que el problema de detección de paráfrasis en textos cortos es un problema que no ha sido resuelto con un carácter definitivo en la actualidad, la inclusión de técnicas basadas en aprendizaje automático supondría un aumento notable en la efectividad de la solución propuesta en el presente Trabajo de Fin de Grado.

7.2. Trabajo a futuro

La detección de paráfrasis es un área de investigación muy recurrente en los últimos años tanto por sus múltiples posibilidades de aplicación como por su gran margen de mejora ligado al desarrollo de las tecnologías de computación.

A causa de esta constante investigación alrededor de las técnicas enfocadas a la detección de paráfrasis y la continua mejora de los recursos disponibles para este fin, las posibilidades en el trabajo a futuro para un método como el expuesto en el presente Trabajo de Fin de Grado son notablemente variadas.

A continuación se describirán las líneas de trabajo a futuro más destacadas para la solución desarrollada en el presente Trabajo de fin de Grado que se han identificado:

En primer lugar y como principal línea de trabajo a futuro se propone añadir técnicas basadas en aprendizaje automático al método expuesto en el presente Trabajo de Fin de Grado. Las técnicas basadas en aprendizaje automático como se ha explicado en el capítulo dos, son capaces de alcanzar una efectividad mayor que las técnicas en las que se basa el método propuesto en el presente Trabajo de Fin de Grado por lo que métodos basados en este tipo de técnicas aportarían más precisión a la solución aquí propuesta.

La efectividad de las técnicas basadas en aprendizaje automático depende directamente del corpus de entrenamiento, ya que en función de éste los algoritmos implementados extraen diversas características propias de la paráfrasis generando así un modelo en el que se basarán para decidir si existe paráfrasis o no en cualquier conjunto de datos que reciban de entrada.

A consecuencia de esta importancia de poseer un corpus completo, que albergue todos los tipos existentes de paráfrasis para conseguir generar un modelo fiable, surge otra línea de trabajo a futuro que consiste en el diseño de un corpus de estas características. El diseño de un corpus como el descrito anteriormente no es una tarea trivial y supondría un gran salto de calidad y efectividad en los métodos de detección de paráfrasis que utilizan técnicas basadas en aprendizaje automático.

En segundo lugar, otra línea a futuro a desarrollar para aumentar las prestaciones de la solución propuesta en el presente Trabajo de Fin de Grado consistiría en la adaptación de la misma para un entorno multilingüe. Para llevarla a cabo es necesario diseñar un corpus completo como el descrito anteriormente con la particularidad que debe añadir información en los idiomas para los que se precisa la detección de paráfrasis.

Finalmente, la última línea a futuro propuesta consiste en adaptar la solución implementada en el presente Trabajo de Fin de Grado para dos aplicaciones concretas: detección de preguntas repetidas en el Congreso de los Diputados y detección de tuits plagiados. Para llevar a cabo la implementación de estas dos aplicaciones, es necesario diseñar un corpus de datos especializado para cada una de las mismas que contengan información representativa de los datos que en el futuro podrán recibir estas aplicaciones.

A estas líneas de trabajo descritas, hay que añadir, con el objetivo de comparar la efectividad de los resultados obtenidos y ser conscientes de la relevancia de los mismos, un análisis de relevancia estadística de los resultados, que proporcionará un criterio fiable para llevar a cabo los objetivos mencionados.

Las líneas a futuro descritas en este capítulo son las principales líneas identificadas con el objetivo de mejorar la solución del presente Trabajo de Fin de Grado. No obstante, la cantidad de ideas para desarrollar y mejorar la solución aquí propuesta es enorme y aumentará más aún conforme avancen las distintas tecnologías de computación.

Anexo A: Resumen en inglés

Introduction

In recent years, the amount of written information in digital format has increased exponentially. This huge volume of information and all the knowledge contained in it, is completely useless if there are no available tools for its management. For this reason, the need to automatically process text has appeared.

To process this information automatically the Natural Language Processing (NLP) [1] has been used. The Natural Language Processing is a branch of Artificial Intelligence, it can be used, for example, to create systems that group documents [4] or classify texts [5]. All these kind of systems use the textual similarity detection as its main asset.

Textual similarity detection is one of the main tasks inside automatic text processing. Its importance comes from the amount of uses it can have, along with easing the detection of similarities in the nuances and complex structures of the language.

Recently, lots of researches and studies in textual similarity detection have been done successfully. Most of them are focused on long texts, where there is sufficient information to reach efficient results, but others are focused on short texts where there is not enough information increasing the difficulty of the task.

Lately, it has been shown that the methods of textual similarity detection that combine several techniques [14] like word comparison [9], word sequence detection [10], word comparison with measures of semantic similarity [11] or techniques based in machine learning, achieve the best results.

In this final degree project, a repeated questions detection problem will be presented and developed. Specifically, it is about a paraphrase detection problem with pairs of questions in english. The problem of this final degree project was inspired from a competition in Kaggle, (https://www.kaggle.com/c/quora-question-pairs), a site that organizes competitions related with machine learning. This problem represents the technological interest of textual similarity detection.

From the competition of Kaggle (www.kaggle.com) the data set of 404.290 pairs of questions in english was obtained, and it was used to develop de solution of the problem showed in this project. This data set is already classified (labeled) so that the question pairs have been identified according to the semantic equivalence of its questions (‘equivalent’ and ‘non equivalent’).

The problem presented in this final degree project consists of deciding in each pair of questions of the data set, if it is made up of two questions with the same mining or not. To solve this problem, the solution method is divided in three parts: pre-processing, evaluation

and decision. It is important to achieve correctly the goal of each of these parts to reach an efficient solution.

The need to process texts automatically and the difficulties of detecting textual semantic similarity, because of the subjectivity and complexity of the language, make the problem presented in this final degree project. Because all of this difficulties, it will be important to combine several techniques to reach an efficient solution.

Currently, the scientific problem of semantic similarity calculation in short texts has not been solved definitively. This fact, poses a reason to investigate and look for an efficient solution to the problem presented in this final degree project.

The Kaggle competition (www.kaggle.com) also poses a reason to investigate an efficient solution to the problem presented, mainly because that platform has offered some benefits that has made possible the implementation of the present project. These benefits are: the labeled data set (404.290 pairs of questions in english), an online development environment and some solution proposals that other competitors published in the platform.

In addition, another reason to investigate the solution to this problem presented in this final degree project was the amount of uses that can be implemented with the algorithms and the ideas of an efficient solution to that problem. Some of these uses are: the detection of repeated questions in ‘ El Congreso de los Diputados’ or the repeated tweets detection to identify plagiarism in Twitter.

The solution presented in this final degree project combines different techniques, some of them have already been implemented and others are implemented in this final degree project for the first time. All of these techniques are focused on paraphrase detection and have been inspired from the previous researches made in this area.

Paraphrase detection

Paraphrase detection has become a very popular objective in recent researches related to semantic similarity detection. Paraphrase is defined as the rewording of something written or spoken.

Paraphrase detection can be applied in many applications like: automatic translation [22], automatic generation of abstracts [23] or identification of plagiarism in texts [24]. However, paraphrase detection also can be the final objective of an application, for example to detect repeated questions in “El Congreso de los Diputados”.

Paraphrase detection among sentence or questions instead of longer texts, has been more frequent among researchers due to the lower computational cost. These researches pose different methods to solve the paraphrase detection problem based in different techniques.

These techniques have evolved over time. At the beginning, the techniques to paraphrase detection were based on lexical similarity [26][27][28], that is to say, these techniques were

based on the amount of identical words in both sentences or its lexical similarity. Then, techniques based on semantic similarity of words [30][31] could be used thanks to tools like WordNet [29], a lexical database in english with which it is possible to estimate a value that represents the semantic similarity among two words.

Recently, techniques based on supervised algorithms relative to machine learning, have been used for paraphrase detection [12][13]. These techniques combined with the techniques based on semantic similarity measures, have achieved the bests results for paraphrase detection [36].

The techniques based on the semantic and lexical similarity of the words try to calculate a score that represent the textual semantic similarity between two sentences or questions. Usually, that score is between “0” and “1”. Where “1” means the maximum value of textual semantic similarity and “0” means the opposite. After getting this value, it is necessary to decide which scores mean paraphrase. Therefore, it is necessary to calculate a threshold and based on it decide which scores mean paraphrase.

The main techniques for paraphrase detection, based on lexical similarity, are the techniques based on the number of words repeated in both sentences or questions. The most important measures are: Dice coefficient [37], Jaccard coefficient [38], Traslape coefficient [39] and cosine coefficient [40].

The techniques for paraphrase detection, based on the semantic similarity of the words, are divided into two groups: knowledge based metrics [74] and corpus based measures [75].

The corpus based measures, get the semantic similarity relationship among words from large amounts of texts. From those texts, it gets information about the semantic similarity of the words, with that information, it is then possible to decide if paraphrase between two sentences exists.

The knowledge based metrics use lexical and semantic information from dictionaries and thesauri to define a semantic comparison criterion among words. The most important tool based in knowledge is WordNet. WordNet is a lexical database in english [29, 30] which main objectives are: to form a simple combination of dictionary and thesaurus and help in the natural language processing.

WordNet is able to calculate a representative value between “0” and “1” of the semantic similarity of a pair of words. That functionality is really useful and it is implemented in WordNet.Similarity, a WordNet package. There are some measures in WordNet.Similarity that can calculate a value that represent the semantic similarity among two words. The most important are: lesk [44], Métrica Leacock y Chodorow [46], Métrica Wu y Palmer [47], Métrica Resnik [48], Métrica Lin[49], Métrica Jiang y Conrath [50], Métrica Path length [51].

The techniques for paraphrase detection based on supervised algorithms relative to machine learning, confront the paraphrase detection as a binary classification problem where the two possible solutions are: “paraphrase” or “no paraphrase”. The algorithms based on these

techniques, are able to create and learn models from data sets already classified. Then, from that created models, the method formed with those algorithms will be able to detect paraphrase in other data sets.

The creation and learning of the model is the most important part of the paraphrase detection process. An scheme that summarizes all the process is shown below:

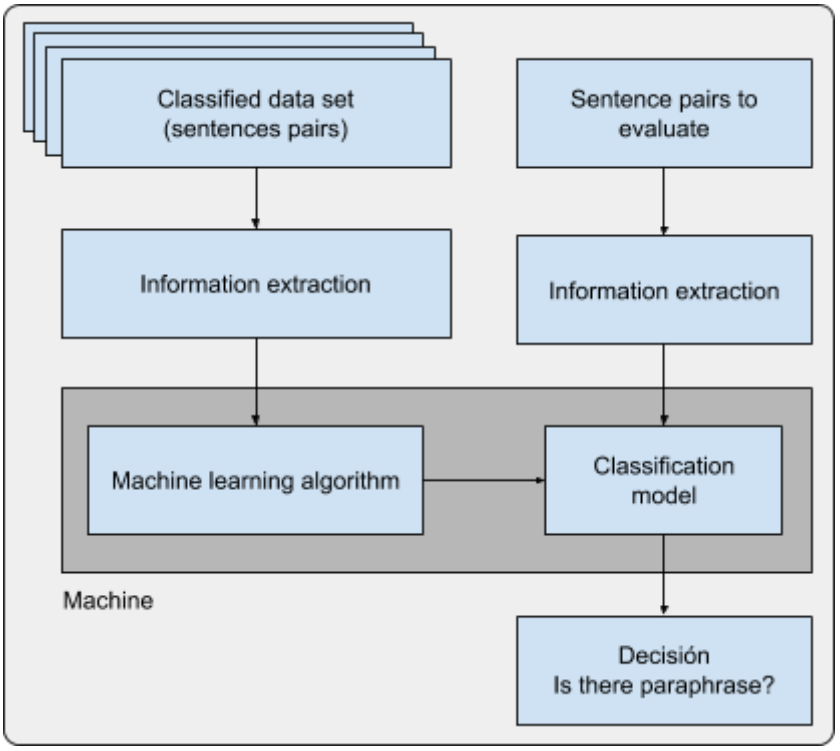

Figure A.1. General scheme of machine learning algorithms operation

These kind of techniques based on supervised algorithms relative to machine learning, are the most effective in paraphrase detection [36].

Problem and solutions

In this final degree project, a problem of paraphrasing detection in pairs of questions is posed. Then, the solution to that problem consists of identifying which pairs of questions have equivalent questions.

Before starting with the solution of the problem, it is important to segment the data set in two sets: training set (270.000 pairs) and test set (134.290). This partition is performed to guarantee the solution quality and avoid the solution overtraining. The training set will be used to implement the solution and the test set to prove its effectiveness.

This solution is divided in three parts: preprocessing, evaluation and decision. This solution process is shown in the scheme below.

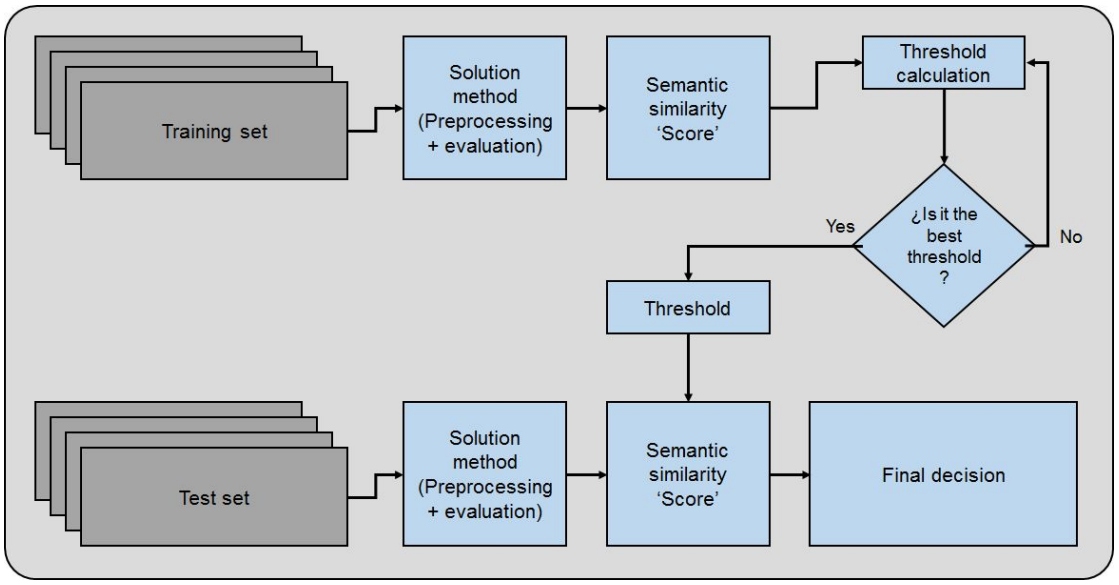

Figure A.2. General scheme of the solution process

Firstly, it is important to adapt the data set to the subsequent needs. A good preprocessing of the data set will greatly facilitate the tasks that come next. So that in this part of the solution, each question of the data set will be divided in tokens (units equivalents to words), all the letters will be converted to lowercase and the punctuation marks, other symbols or words ("stop words") that do not give semantic information, will be eliminated. The remaining tokens will be lemmatized, that is, converted into its lemma.

Secondly, it is important to evaluate the preprocessed data set to calculate the score that represents the textual semantic similarity of each pair of questions. This score will be a number between "0" (minimum value of similarity) and "1" (maximum value of similarity). To reach an efficient solution to the posed problem, five methods based on different techniques have been implemented. Each of these methods will give one value that represent the textual semantic similarity for each pair of questions. To get just one final value that represent the textual semantic similarity of each pair of questions, the average will be performed with the first four methods. The fifth method combines the techniques used in the first four methods so, to choose the final value, the solution of the fifth method and the mean of the first four methods solution will be compared.

Method 1. Literal word comparator.

This method compares the words of both questions of each pair and gets a score based on the number of identical words that appear in both questions. To get that score normalized, the following measures are used: Dice coefficient [37], Jaccard coefficient [38], Traslape coefficient [39] and cosine coefficient [40].
Moreover, to get just one score per pair of questions, the average between the four metrics is performed.

Method 2. Counting method

This method is also based on the number of repeated words in each pair of questions to get a score that represent its textual semantic similarity. This method has a particular way to calculate called *"Counting method"* [68].

To get a score between "0" and "1", that represent the textual semantic similarity of the each pair of questions, it is necessary to normalize the value got from the number of words that appear in both questions. That normalization factor is the product of the number of different words in both questions, multiplied by the number of words of the shorter question.

Method 3. Semantic word comparator.

This method is based on the semantic similarity of the question words of each pair. To get the score that represent the textual semantic similarity of each pair of questions, a comparison between every word of the question 1 and every word of the question 2 is performed, following this procedure, a similarity matrix between the words of both questions is performed.

Moreover, to normalize the score, the resulting value is divided by the square root of the product of two similarity matrix performed with the words of the same question (one of them with the words of the question 1 and the other one with the words of question 2):

$$S_{SN}(p_1, p_2) = \frac{S_{SEM}(p_1, p_2)}{\sqrt{S_{SEM}(p_1, p_1) \times S_{SEM}(p_2, p_2)}} \quad \text{(A.1.)}$$

Where $S_{SEM}(p_1, p_2)$ is the function that returns the similarity matrix between p_1 and p_2 (question 1 and question 2) and $S_{SN}(p_1, p_2)$ is the function that return the semantic similarity value of p_1 y p_2.

To perform this operation, it is necessary to have at least one semantic similarity measure to get a representative value of the semantic similarity between the words of both questions of each pair. In this method, four similarity measures have been used (Jiang y Conrath, Path length y Wu y Palmer from WordNet tool and other one with Word2Vec tool).

To get just one score, an average between the four values obtained from the four different semantic similarity measures is performed.

This method is based on the research of Fernando and Stevenson [31] who implemented the similarity matrix to paraphrase detection for the first time.

Method 4. Maximum similarity

This method, as method 3, is based on the semantic similarity of the question words of the pairs. In this case, to get a value that represents the semantic similarity of each pair, just one similarity measure per word has been taken into consideration. That is to say, the semantic similarity calculation between all the words of one question with the words of the other question is performed . Furthemore, for each word the biggest semantic similarity value is chosen.

To get one value that represents the semantic similarity of each pair of questions, all the values corresponding with the words of each pair are added. Then, it is necessary to normalize that value, so that, this value is divided by the sum of the number of words of each question of the pair.

$$S_{msN}(p_1, p_2) = \frac{S_{ms}(p_1, p_2)}{|p_1| + |p_2|} \quad \text{(A.2.)}$$

Where $p_1 y\ p_2$ are the questions 1 and question 2, $S_{msN}(p_1, p_2)$ is the function that returns the semantic similarity without the normalization and $|p_1|$ y $|p_2|$ are the number of words in $p_1 y\ p_2$ respectively.

To perform this operation, as in the method 3, it is necessary to have at least one semantic similarity measure to get a representative value of the semantic similarity between the words of both questions of each pair. In this method, four similarity measures (Jiang y Conrath, Path length y Wu y Palmer from WordNet tool and other one with Word2Vec tool) have been used.

To get just one score, the average between the four values obtained from the four different semantic similarity measures has to be performed .

This method is based on the method implemented by Mihalcea, R., Corley, C. & Strapparava, C. [52].

Method 5. Hybrid method

This method combines semantic similarity techniques and lexical similarity to get a representative value of the semantic similarity among the questions of each pair. This method also includes a semantic dissimilarity measure that decreases the semantic similarity value of a pair of questions if the semantic similarity value of two words is smaller than a threshold value.

To get the value that represents the semantic similarity among the questions of each pair, in this method the techniques used in the method 1 and method 4 are combined. In addition, in the part corresponding to the method 4, when the semantic similarity value of any word is smaller than 0.5, that value is subtracted instead of being added to the score.

To get just one value, the average between the value obtained from the semantic similarity techniques and the value obtained from the lexical similarity techniques is performed.

Finally, it is important to decide, based on the values obtained from the methods explained, in which pairs of questions paraphrase exists. To decide it, it is important to calculate a threshold value and, based on its value decide if there is any paraphrase. Therefore, if the value that represents the semantic similarity of both questions of each pair is bigger than the threshold value, it will be decided that paraphrase exists in that pair.

The calculation of the threshold value is very important because from this value depends the final effectiveness of this final degree project solution. So that, to calculate the best threshold for the values obtained in the methods implemented, an iteration with some threshold values will be made. Then the threshold value that returns the best result, will be the threshold value chosen to decide.

Experiments and results

The methods explained above have been tested with the same process as those described in the general scheme of the solution process in the *Figure A.2.*

The results obtained from those methods are shown and explained below:

Method	Accuracy	F measure
1. Literal word comparator	0.676	0.265
2. Counting method	0.659	0.246
3. Semantic word comparator	0.689	0.277
4. Maximum similarity	0.691	0.279
5. Hybrid method	0.706	0.286
6. Average (methods 1-4)	0.694	0.282

Chart A.1. Accuracy and F measure for the methods implemented in this final degree project.

The obtained results in the experiments performed in the Method 1. Literal word comparator and in the Method 2. Counting method, show that these methods have a good accuracy and processing time ratio, taking into account that the processing speed of these methods is four times faster than the other methods processing speed. These results also show that the Method 1 has a better performance than Method 2. It is because the combination of several techniques in one single method increase the method precision.

The obtained results in the experiments performed in the Method 3. Semantic word comparator and Method 4. Maximum similarity, based on semantic similarity techniques, show better performance than the methods based on words repetition. The explanation for this is that these methods (Method 3 and Method 4) take into account the semantic words similarity in addition to the word repetition.

These obtained results, also show that the performance of Method 4 is better than the performance of Method 3 but both are very similar. Those similar results are caused by the preprocessing part. This part decreases considerably the number of words per question so that the different characteristics among both methods do not have as much repercussion in the final result.

The obtained results in the experiments performed in the Method 5. hybrid method, and the average performed among the first four methods, show that the semantic dissimilarity measure works efficiently. The Hybrid Method present better results than the average performed among the first four methods.

Because of this better performance of the Hybrid Method, it has been chosen as the solution to the presented problem in this final degree project.

Conclusions

The obtained results show that the accuracy of the developed methods increases as the number of different techniques used increases. It is because as the number of used techniques increase, the amount of information that can be stored to detect paraphrase also increase.

However, the accuracy of all the methods presented in this final degree project is similar and does not suppose a definitive solution for the posed problem. It is true that currently the scientific problem of semantic similarity calculation in short texts has not been solved definitively. The combination of techniques based on machine learning with the techniques developed in this final degree project would mean an increase in the effectiveness of the solution.

Bibliografía

[1] Christopher D Manning & Hinrich Schutze. Foundations of statistical natural language processing. MIT press, 1999.

[2] V Melissa Holland, Michelle R Sams, & Jonathan D Kaplan. (2013). Intelligent language tutors: Theory shaping technology. Routledge.

[3] Karin Verspoor & Kevin Bretonnel Cohen. (2013). Natural language processing. Encyclopedia of Systems Biology, páginas 1495-1498.

[4] Wui Lee Chang, Kai Meng Tay, & Chee Peng Lim. (2014). A new evolving tree for text document clustering and visualization. In Soft Computing in Industrial Applications, páginas 141-151. Springer.

[5] Weiyuan Li & Hua Xu. (2014) Text-based emotion classification using emotion cause extraction. Expert Systems with Applications, 41(4):1742-1749.

[6] Merin Francis & Ramachandran Nair KN. (2014). An algorithm for plagiarism detection in malayalam language documents using modified n-gram model.

[7] Lei Wu, Steven CH Hoi, & Nenghai Yu. (2010). Semantics-preserving bag-of-words models and applications. Image Processing, IEEE Transactions on, 19(7):1908–1920.

[8] Xiaoying Liu, Yiming Zhou, & Ruoshi Zheng. (2007). Sentence similarity based on dynamic time warping. In Semantic Computing, 2007. ICSC 2007. International Conference on, páginas 250-256. IEEE.

[9] Bo Han & Timothy Baldwin. (2011). Lexical normalisation of short text messages: Makn sens a # twitter. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, páginas 368–378. Association for Computational Linguistics.

[10] Tristan Miller, Chris Biemann, Torsten Zesch, & Iryna Gurevych. (2012). Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In COLING, páginas 1781–1796.

[11] Frane Saríc, Goran Glavas, Mladen Karan, Jan Snajder, & Bojana Dalbelo Basic. (2012). Takelab: Systems for measuring semantic text similarity. In Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, páginas 441–448. Association for Computational Linguistics.

[12] Kozareva, Z. & Montoyo, A. (2006). Paraphrase identification on the basis of supervised machine learning techniques. Lecture Notes in Computer Science 4139, pp 524-533.

[13] Madnani, N., Tetreault, J. & Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. Proceedings 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 182-190). Montréal, Canada.

[14] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, & Weiwei Guo. (2013). sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics. Citeseer.

[15] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, & Murat Demirbas. (2010) Short text classification in twitter to improve information filtering. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, páginas 841–842. ACM.

[16] Julio Castillo & Paula Estrella. Semantic textual similarity for mt evaluation. (2012). In Proceedings of the Seventh Workshop on Statistical Machine Translation, páginas 52–58. Association for Computational Linguistics.

[17] Wui Lee Chang, Kai Meng Tay, & Chee Peng Lim. (2014). A new evolving tree for text document clustering and visualization. In Soft Computing in Industrial Applications, páginas 141-151. Springer.

[18] Jesús Oliva, José Ignacio Serrano, María Dolores del Castillo, & Ángel Iglesias. (2011) Symss: A syntax-based measure for short-text semantic similarity. Data & Knowledge Engineering, 70(4):390-405.

[19] Rahul Bhagat & Eduard Hovy. (2013). What is a paraphrase?

[20] Michael Mohler, Razvan C Bunescu, & Rada Mihalcea. (2011). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In ACL, páginas 752-762.

[21] Paráfrasis. (2018). En Diccionario de la lengua española. Recuperado de http://www.rae.es/

[22] Zhou, L., Lin, C. & Hovy, E. (2006). Re-evaluating machine translation results with paraphrase support. Proceedings 2006 conference on empirical methods in natural language processing (pp. 77-84). Sydney, Australia.

[23] Barzilay, R. & McKeown, K. (2005). Sentence fusion for multidocument news summarization. Computational Linguistics 31(3), pp. 297-328.

[24] Barrón, A. et al. (2013). Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. Computational Linguistics 39(4), pp. 917-947.

[25] Rinaldi, F. et al. (2003). Exploiting paraphrases in a question answering system. Proceedings second international workshop on paraphrasing (pp. 25-32). Sapporo, Japan.

[26] Clough, P. et al. (2002). METER: MEasuring TExt Reuse. Proceedings 40th Annual Meeting on Association for Computational Linguistics (pp. 152-159). Morristown, USA.

[27] Zhang, Y. & Patrick, J. (2005). Paraphrase identification by text canonicalization. Proceedings Australasian language technology workshop (pp. 160-166). Sydney, Australia.

[28] Qiu, L., Kan, M. & Chua, T. (2006). Paraphrase recognition via dissimilarity significance classification. Proceedings 2006 Conference on Empirical Methods in Natural Language Processing (pp. 18-26). Sydney, Australia.

[29] Pedersen, T., Patwardhan, S. & Michelizzi, J. (2004). WordNet::Similarity - Measuring the relatedness of concepts. Proceedings HLT-NAACL--Demonstrations '04 Demonstration Papers at HLT-NAACL (pp. 38-41). Boston, USA.

[30] Corley, C. & Mihalcea, R. (2005). Measuring the semantic similarity of texts. Proceedings ACL workshop on empirical modeling of semantic equivalence and entailment (pp. 13-18). Michigan, USA.

[31] Fernando, S. & Stevenson, M. (2008). A semantic similarity approach to paraphrase detection. Proceedings 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics (pp. 45-52). Oxford, England.

[32] Malakasiotis, P. (2009). Paraphrase recognition using machine learning to combine similarity measures. Proceedings ACL-IJCNLP 2009 Student Research Workshop (pp. 27-35). Suntec, Singapore.

[33] Das, D. & Smith, N. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. Proceedings 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (pp. 468-476). Singapore.

[34] Cordeiro, J., Dias, G. & Brazdil, P. (2007). A metric for paraphrase detection. Proceedings International Multi-Conference on Computing in the Global Information Technology (pp. 7-16). Barcelona, Spain.

[35] Cordeiro, J., Dias, G. & Brazdil, P. (2007). New functions for unsupervised asymmetrical paraphrase detection. Journal of Software 2(4), pp. 12-23.

[36] Yusdanis Feus P. & Isvani Frías B. (2015). Estado actual de la detección de paráfrasis. RACCIS 5(2), pp. 25-36.

[37] Mohammad Yahya H Al-Shamri. (2014). Power coefficient as a similarity measure for memory based collaborative recommender systems. Expert Systems with Applications.

[38] Huang Cheng-Hui, Yin Jian, & Hou Fang. (2011). A text similarity measurement combining word semantic information with tf-idf method. Chinese journal of computers, 34(5): 856-864.

[39] Wael H Gomaa & Aly A Fahmy. (2013). A survey of text similarity approaches. International Journal of Computer Applications, 68(13):13-18.

[40] Álvarez Carmona, Miguel Ángel. (2014). Detección de similitud semántica en textos cortos. Instituto Nacional de Astrofísica, Óptica y Electrónica. Tonantzintla, Puebla.

[41] Dolan, B., Quirk, C. & Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. Proceedings 20th international conference on Computational Linguistics (pp. 350). Morristown, USA.

[42] Miller, G. et al. (1990). Introduction to WordNet: An on-line lexical database. International journal of lexicography 3(4), pp. 235-244.

[43] Miller, G. & Fellbaum, C. (1998). WordNet: An electronic lexical database. WordNet: An electronic lexical database. Cambridge: MIT Press.

[44] Michael Lesk. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In Proceedings of the 5th annual international conference on Systems documentation, páginas 24–26. ACM.

[45] Banerjee, S. & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. Proceedings Eighteenth International Joint Conference on Artificial Intelligence (pp. 805-810). Acapulco, Mexico.

[46] Claudia Leacock & Martin Chodorow. (1998). Combining local context and wordnet similarity for word sense identification. WordNet: An electronic lexical database, 49(2):265-283.

[47] Wu, Z. & Palmer, M. (1994). Verbs semantics and lexical selection. Proceedings 32nd annual meeting on Association for Computational Linguistics (pp. 133-138). New Mexico, USA.

[48] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. Proceedings International Joint Conference on Artificial Intelligence, (pp. 448-453). Montreal, Canada.

[49] Lin, D. (1998). An information-theoretic definition of similarity. Proceedings Fifteenth International Conference on Machine Learning (pp. 296-304). Madison, USA.

[50] Jiang, J. & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings International Conference on Research in Computational Linguistics (pp. 1-15). Taiwan, China.

[51] Alexander Budanitsky & Graeme Hirst. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics, 32(1):13-47.

[52] Mihalcea, R., Corley, C. & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. Proceedings American Association for Artificial Intelligence (pp. 775-780). Boston, USA.

[53] Spärck, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of documentation 28(1), pp. 11-21.

[54] Clear, J. (1993). The British National Corpus. In Landow, G. & Delany, P. (Eds.), The digital word: text-based computing in the humanities (pp. 163-187). Cambridge, MIT Press.

[55] Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. Lecture Notes in Computer Science 2167, pp. 491-502.

[56] Thomas K Landauer, Peter W Foltz, & Darrell Laham. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3):259-284.

[57] Stevenson, M. & Greenwood, M. (2005). A semantic approach to IE pattern induction. Proceedings 43rd Annual Meeting on Association for Computational Linguistics (pp. 379-386). Morristown, USA.

[58] Charniak, E. (2000). A maximum-entropy-inspired parser. Proceedings 1st North American chapter of the Association for Computational Linguistics conference (pp. 132-139). Seattle, USA.

[59] Pradhan, S. et al. (2004). Shallow Semantic Parsing using Support Vector Machines. Proceedings Human Language Technology Conference/North American chapter of the Association of Computational Linguistics (pp. 233-240). Boston, USA.

[60] Papineni, K. et al. (2002). BLEU: A method for automatic evaluation of machine translation. Proceedings 40th annual meeting on association for computational linguistics (pp. 311-318). Morristown, USA.

[61] Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. Proceedings second international conference on Human Language Technology Research (pp. 138-145). San Diego, USA.

[62] Snover, M. et al. (2006). A study of translation edit rate with targeted human annotation. Proceedings 7th Conference of the Association for Machine Translation in the Americas (pp. 223-231). Cambridge, USA.

[63] Snover, M. et al. (2009). TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. Machine Translation 23(2-3), pp. 117-127.

[64] Denkowski, M. & Lavie, A. (2010). Extending the METEOR machine translation evaluation metric to the phrase level. Proceedings 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 250-253). Stroudsburg, USA.

[65] Habash, N. & Elkholy, A. (2008). SEPIA: Surface span extension to syntactic dependency precision-based MT evaluation. Proceedings Workshop on Metrics for Machine Translation at AMTA (pp. 1-3). Berlin, Germany.

[66] Chan, Y. & Ng, H. (2008). MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation. Proceedings 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 55-62). Columbus, USA.

[67] Fernando, S. (2007). Paraphrase Identification. Master's thesis. University of Sheffield, UK

[68] Estévez, Macarena. (2016). Cómo medir la similitud entre textos. Inteligencia analítica. Recuperado de: https://inteligencia-analitica.com/como-medir-similitud-entre-textos/.

[69] The Python Software Foundation. (2018). History and License. Recuperado de: https://docs.python.org/3/license.html.

[70] Open source initiative. (2018). Recuperado de: https://opensource.org/.

[71] Free software foundation. (2018). Categorías de software libre y software que no es libre. Recuperado de: http://www.gnu.org/philosophy/categories.es.html#CopyleftedSoftware.

[72] The Python Software Foundation. (2001-2018). PEP 8 -- Style Guide for Python Code. Recuperado de: https://www.python.org/dev/peps/pep-0008/.

[73] Tu salario. (2018). Recuperado de: https://tusalario.es/.

[74] Juan Antonio Prieto Velasco. (2013). A corpus-based approach to the multimodal analysis of specialized knowledge. Language resources and evaluation, 47(2):399-423.

[75] Carmen Banea, Yoonjung Choi, Lingjia Deng, Samer Hassan, Michael Mohler, Bishan Yang, Claire Cardie, Rada Mihalcea, & Janyce Wiebe. (2013). Cpn-core: A text semantic similarity system infused with opinion knowledge. Atlanta, Georgia, USA, página 221.